الوصول الى الواقعية

- الفصل الأول: مقدمة
- الفصل الثاني: خط الإنطلاق
- الفصل الثالث: كنّ موجز
- الفصل الرابع: المهم فالأهم
- الفصل الخامس: إختيار الخصائص
- الفصل السادس: بناء التطبيق
- الفصل السابع: التنظيم
- الفصل الثامن: التوظيف
- الفصل التاسع: تصميم البرنامج
- الفصل العاشر: الأكواد
- الفصل الحادي عشر: إختيار الكلمات
- الفصل الثاني عشر: الأسعار و الإشتراكات
- الفصل الثالث عشر: الترويج
- الفصل الرابع عشر: الدعم
- الفصل الخامس عشر: قبل الإنطلاق
- الفصل السادس عشر: الختام

الفصل الأول: مقدمة

ما المقصود بالوصول إلى الواقعية ؟

هل ترغب في بناء تطبيق ويب ناجح؟ إذا حان وقت الوصول إلى الواقعية. الوصول إلى الواقعية أفضل وأسرع وأبسط طريقة لبناء البرمجيات.

- الوصول إلى الواقعية هو تخطي الأمور غير الواقعية (الرسومات البيانية، الأسهم، الصناديق، المخططات وغيرها) لصالح بناء ماهو واقعي.
- الوصول إلى الواقعية تعني الأقل. الأقل في الحجم ،الأقل في البرمجة ،الأقل في الخصائص ،الأقل في العمل الورقي. باختصار الأقل في كل شيء غير ضروري (معظم ما تظنه ضرورياً في الحقيقة هو ليس كذلك).
- الوصول إلى الواقعية هي بقاء التطبيق صغيراً ونشيطاً .
- الوصول إلى الواقعية يعني أن تبدأ بالواجهة، أي بالشاشات التي سيستخدمها العملاء. فهي تنطلق من الخبرة الحقيقية للعميل متجهه للداخل. هذا يجعلك تحصل على الواجهة الصحيحة قبل حصولك على البرمجية الخاطئة.
- الوصول إلى الواقعية يدور حول تكرار العمليات وتخفيض تكلفة التغيير . الوصول للواقعية تعني إطلاق المنتج، تعديله وتطويره، و الاستمرار على ذلك، وهذا ما يجعل من الوصول إلى الواقعية أفضل الطرق لبناء تطبيقات الويب.
- الوصول إلى الواقعية هي إيصال ما يريده العملاء فقط ،وتجاهل ما لا يحتاجونه.

فوائد الوصول إلى الواقعية

الوصول إلى الواقعية تقدم أفضل النتائج لأنها تُجبرك على التعامل مع المشاكل نفسها لا مع الأفكار التي تدور حولها. فهي تُجبرك على التعامل مع الحقيقة.

الوصول إلى الواقعية هي تجنب تقارير المواصفات الداخلية functional specs وغيرها من توثيقات مؤقتة، وذلك من أجل بناء شاشات واقعية. تقارير المواصفات الداخلية أمور شكلية واتفاقيات وهمية، في حين تعتبر صفحة الويب هي الواقعية. لأن ما سوف يراه ويستخدمه العملاء هو الأهم. الوصول للواقعية تعني الوصول سريعاً ، وهذا يعود إلى أن القرارات البرمجية يتم اتخاذها بناء على أمور واقعية لا على أفكار مجردة.

أخيراً، الوصول إلى الواقعية هو المنهج الأمثل لبرمجيات الويب. تقديم برمجيات معلبة ومن ثم الانتظار سنة أو أكثر لتقديم التحديثات هو نموذج تقليدي لم يعد موجوداً . بخلاف برمجيات سطح المكتب –البرمجيات التي تحتاج إلى تثبيت- فإن تطبيقات الويب يتم تطويرها بشكل يومي. الوصول إلى الواقعية يزيد من كفاءتك في التعامل مع كافة جوانب هذه الميزة.

كيف تكتب برمجيات قوية

الكتابة القوية مختصرة. فأي جملة يجب ألا تحتوي على كلمات غير ضرورية. وأي فقرة يجب ألا تحتوي جمل غير ضرورية. كذلك التصاميم يجب ألا تحتوي على خطوط غير ضرورية .والأجهزة يجب ألا تحتوي على قطع غير ضرورية. هذا لا يعني أن يجعل الكاتب كل الجمل قصيرة أو يتفادى كل التفاصيل و يتعامل مع المواضيع بإيجاز مفرط، بل عليه أن يستخدم كل ما هو مفيد .

—المصدر "نمط العناصر" بواسطة ويليم سترنك جي أر

لا مزيد من التضخم

المنهج القديم قائم على عمليات مطولة وبيروقراطية و عمل شيء عن شيء أخر لحماية المصالح. والنتيجة الحتمية هي برمجيات متضخمة ومعرضة للنسيان بسبب تدني الجودة.

الوصول إلى الواقعية هو التخلص من...

- الجداول الزمنية التي تمتد لشهور أو ربما سنوات
- تقارير المواصفات الداخلية غير الواقعية
- النقاشات الطويلة
- اجتماعات الموظفين المطوّلة
- "الحاجة" لتوظيف الكثير من الموظفين
- أرقام الإصدارات التي لا معنى لها
- الخرائط البدائية والتي تتنبأ بمستقبل باهر
- الخيارات اللا نهائية من التفضيلات
- إسناد المهام إلى فرق دعم خارجية
- اختبارات المستخدم غير الواقعية
- الأوراق غير المجدية
- التدرج الهرمي من الأعلى للأسفل

لست بحاجة إلى فريق كبير أو مبالغ ضخمة أو دورة تطوير برمجيات طويلة لبناء برنامج رائع . هذه الأشياء هي مكونات التطبيقات الجامدة، البطيئة والمعقدة. الوصول إلى الواقعية يسلك المسار الأخر.

في هذا الكتاب سنوضح لك...

- أهمية وجود فلسفة
- لماذا الأقل هو الأفضل
- كيف تبني الأقل

- كيف تنتقل بسرعة من الفكرة إلى الواقع
- كيف توظف فريقك
- لماذا يجب عليك التصميم من الداخل للخارج
- لماذا الكتابة مهمة جداً
- ماذا يجب أن تعمل أقل من منافسيك
- كيف تروج لتطبيقك و تنشره للعالم
- أسرار الدعم الفني الناجح
- نصائح تبقيك في حالة استمرارية بعد الانطلاق
- وغيرها ....

سينصب التركيز على الأفكار بشكل عام. لن نشغلك بتفاصيل الأكواد أو بخدع الأنماط CSS . سنركز على الأفكار والفلسفات الرئيسية التي تقوم عليها عمليات "الوصول إلى الواقعية".

هل الكتاب مناسب لي؟

إذا كنت رجل أعمال، مصمم، مبرمج أو مسوق تعمل على فكرة ضخمة.

القواعد القديمة لم تعد صالحه. هل ستوزع برنامجك على أقراص مدمجة كل عام؟ كيف سيكون رقم الإصدار لنسخة عام 2002؟ هذه الأمور غير مهمة. كل ما تحتاجه البناء، الإطلاق، التعديل والتطوير. ومن ثم تكرار هذه العمليات مرات ومرات.

قد تكون من من لم يسبق له التعامل مع التطوير السريع agile development أو الهياكل التجارية business structures ، لكنك متلهف لمعرفة المزيد.

إذا كنت كذلك، فان هذا الكتاب موجهه لك..

ملاحظة: هذا الكتاب يركز على بناء تطبيقات ويب ، لكن الكثير من الأفكار الواردة فيه قابلة للتطبيق في الأنشطة الأخرى البعيدة عن البرمجة. الاقتراحات الواردة في الكتاب مثل الفرق الصغيرة، النماذج الأولية السريعة وغيرها تستطيع استخدامها عند البدء في عمل تجاري أو عند كتابة كتاب أو عند تصميم موقع أو تسجيل ألبوم وغيرها من أعمال. فور ما تطبق "الوصول إلى الواقعية" في أحد مناحي الحياة، ستجد أن ما فيه من مفاهيم قابلة للتطبيق على عدد كبير من الأنشطة.

حول 37signals

ماذا نقدم

37signals فريق عمل صغير يقدم برمجيات بسيطة ومركزة. منتجاتنا تبقيك منظماً وتساعدك على التعاون مع الآخرين. أكثر من 350000 شخص ومؤسسة صغيرة تستخدم تطبيقاتنا لإنجاز مهامها . كتب جيرمي وجستف من صحيفة وول ستريت" إن منتجات 37signals بسيطة بشكل جميل و أنيقة و أدواتها سهلة الفهم إلى درجة أن شاشة برنامج الآوت لوك ستبدو نوعاً من العذاب مقارنة بها " . تطبيقاتنا لا تضعك في موقف صعب

طريقتنا الخاصة في العمل

نؤمن بأن البرمجيات معقدة جداً . فهي تحتوي على الكثير من المميزات والخيارات وبالتالي تحتاج إلى جهد كبير لتعلمها. في منتجاتنا نتعمد تقديم منتجات أقل من منافسينا. فنحن نبني منتجات تعمل بشكل أذكى و أسرع. منتجات تتسم بسهولة الاستخدام وتسمح لك بعمل الأشياء بالطريقة التي تريدها.

منتجاتنا

في تاريخ نشر هذا الكتاب، فإننا نملك خمسة منتجات تجارية بالإضافة إلى إطار عمل واحد مفتوح المصدر.

Basecamp قلب النظرة تجاه إدارة المشاريع. فبدلاً من خرائط جانت و الرسوم البيانية و الجداول الحسابية المتخمة بالبيانات الإحصائية، فإن Basecamp يوفر منتدى للحوار، قوائم المهام اليومية، جدولة يومية، كتابة تعاونية و مشاركة الملفات. وحتى هذه اللحظة اتفق مئات الآلاف على أنه أفضل طريقة للتواصل في المشاريع. قال فارهد مانجو من Salon.com " تمثل Basecamp مستقبل البرمجيات على شبكة الانترنت" .

Campfire قدم لمجال الأعمال التجارية خدمة المحادثة الجماعية وبشكل بسيط . يدرك قطاع الأعمال مدى أهمية المحادثة الجماعية . المحادثات التقليدية رائعة عندما تكون المحادثة بين شخصين ولكن عندما يكون العدد أكبر من ذلك فهي سيئة. Campfire يحل هذه المشكلة وغيرها من المشاكل الأخرى.

Backpack هو بديل لبرامج إدارة المعلومات الشخصية المربكة و المعقدة و التي تحمل شعارات مثل"نظم حياتك في 25 خطوة بسيطة" . Backpack يتيح لك وببساطة عمل الصفحات، تسجيل الملاحظات، تدوين المهام، التذكير بالمواعيد من خلال البريد الإلكتروني أو الهاتف النقال والأخيرة هي خدمة مبتكرة في هذا المجال ننفرد بتقديمها لفئة في أمس الحاجة لها. توماس ويبر من صحيفة وول ستريت قال انه أفضل منتج في مجاله، في حين أن ديفيد بوقي من نيورك تايمز أطلق عليه "أروع أداة تنظيم" .

Writeboard يتيح لك الكتابة ، المشاركة ، التنقيح و مقارنة النص منفرداً أو مع نصوص أخرى. أنه البديل الأحدث لمعالجات النصوص المتضخمة بالمييزات والتي لا تحتاج 95% منها أثناء عملية الكتابة . قال جون قروبر من Daring Fireball: "قد يكون Writeboard أوضح و أبسط تطبيق ويب رأيته في حياتي" . ويقول الأب الروحي للويب جيفيري زيلدمان " العقول الرائعة في 37signals فعلت ذلك مجدداً ."

Ta-da List تساعدك على الاحتفاظ بقوائم المهام الخاصة بك مع بعضها البعض وبشكل منظم. في Ta-da List لك الخيار في مشاركة تلك القوائم مع الآخرين وبسهولة أو الاحتفاظ بها لنفسك. لا يوجد طريقة تساعدك في إنجاز المهام أسرع من ذلك. أكثر من 100000 قائمة بما يقارب 1000000 مهمة تم إنشاؤها حتى هذه اللحظة.

Ruby on Rails للمطورين، هي مجموعة كاملة و إطار عمل مفتوح المصدر معتمد على لغة الروبي يتيح لك كتابة تطبيقات واقعية بسرعة و سهولة. إطار العمل هذا يراعي انشغال الأفراد، لذا فهو يجعل المبرمج يركز فقط على فكرته. يقول نثان توركنقتون من إمبراطورية النشر O'Reilly: " روبي أون ريلز مُذهلة بحق، استخدامها يشبه مشاهدة فلم كنج-فو ، بينما العشرات من أطر العمل السيئة تستعد لاستقبال الوافدين الجدد فقط من أجل أن تضربهم بكل ما هو سيء و بشتى الطرق الخلاقة" .

يمكنك قراءة المزيد حول شركتنا ومنتجاتنا على الموقع التالي www.37signals.com.

التنبيهات، إخلاء المسئولية، وغيرها من إجراءات وقائية

لتكون على بينة، هذه ردونا على بعض الشكاوى التي تصلنا بين الحين والآخر :

"هذه التقنيات لن تعمل معي"

الوصول إلى الواقعية نظام لم يعمل معنا بشكل جيد. سبق أن قلنا أن الأفكار الواردة في هذا الكتاب لا يمكن تطبيقها على كل المشاريع. إذا كنت تسعى لبناء نظام أسلحة، مصنع يدار بالطاقة النووية ، نظام بنكي لملايين العملاء أو نظام حياتي/مالي يجب أن تتوقف عن تطبيق بعض ما جاء في الكتاب.انطلق وكن محتاطاً بالمزيد من الإجراءات الوقائية.

لا يعني هذا أنك إما أن تطبق كل ما في الكتاب من مقترحات أو لا تطبق شيئاً على الإطلاق. حتى ولو لم تقم بتطبيق الوصول إلى الواقعية بشكل كلي، من المحتمل أن تكون هناك بعض الأفكار التي يمكنك التسلل من خلالها لتحقيق المطلوب.

"أنتم لم تخترعوا تلك الفكرة"

نحن لا ندعي اختراع هذه التقنيات. فالكثير من المفاهيم كانت موجودة بشكلٍ أو بأخر ولمدة طويلة. لا تغضب عندما تقرأ نصائح تذكرك في شيء سبق أن قرأته في مدونة أو حتى في كتاب نشر قبل 20 سنة. هذه الأمور محتمل وجوده في كل مكان. التقنيات الواردة ليست حصرية لفريق 37signals. فقط نحن نخبر الآخرين كيف نعمل وما الأمور التي نجحت معنا.

"انتم تتخذون نظرة بيضاء وسوداء أكثر من اللازم"

إذا كنا نبدو كمدعي المعرفة بكل الأمور، لا تتسرع في الحكم علينا . نحن نؤمن بأن من الأفضل أن يتم تقديم الأفكار بشكلٍ واضح بدلاً من التغريد خارج السرب. إذا كان هذا يبدو كنوع من أنواع الاعتداد بالنفس أو الغرور، اعتبره كذلك. نفضل أن نكون مغامرين على أن نضعف من شأن الأمور من خلال قول "أنه يتوقف على.." بالطبع سيكون هناك مواقف تحتاج فيها لشد أو كسر ما نذكره من قواعد. فبعض التكتيكات قد لا تنطبق على حالتك. لذا استخدم حنكتك و خيالك.

"لن يعمل هذا الشيء في شركتي"

تعتقد أنك أكبر من أن تطبق الوصول إلى الواقعية؟ حتى شركة مايكروسوفت تطبق الوصول إلى الواقعية (و نحن نشك أنك أكبر منهم).

حتى ولو كانت شركتك تعمل بجداول زمنية طويلة ومن خلال فرق عمل كبيرة ، فمازال هناك طرق تمكنك من الوصول إلى الواقعية. أول خطوة قم بتقسيم العمل إلى وحدات صغيرة. لأنه عندما يكون هناك عدد كبير من الأشخاص ستنخفض الإنتاجية. الأقل يعني الأسرع و الأفضل في انجاز المهام.

قد يبدو ما سنقوله كنوع من الترويج، لكن هو في الحقيقة ما نؤمن به. حاول أن تجعل شركتك تتبع عمليات الوصول إلى الواقعية. فأجعلهم يقرأون هذا الكتاب، ويتعرفوا على النتائج الواقعية التي سيحصلون عليها في وقت أقل وفريق عمل أصغر.

وضح لهم أن الوصول إلى الواقعية هي طريقة لا تحتاج إلى مجازفة كبيرة أو إلى استثمارات ضخمة لاختبار المفاهيم الجديدة. يمكنك أن تبرهن لهم ذلك من خلال إقامة مشروع صغير داخل المشروع الأم.

أو إذا أردت أن تكون أكثر شجاعة حاول أن تعمل في الخفاء بعيداً عن الأنظار ثم أعرض النتائج. هذا الطريقة سلكها فريق Start.com في محاولتهم للوصول إلى الواقعية في شركة مايكروسوفت. يقول روبرت سكوبل كاتب تقني في مايكروسوفت " لقد شاهدت فريق Start.com يعملون، لم يطلبوا الإذن" . "لديهم مرشد يساعدهم. و هم يستقطعون جزء قليل من الوقت لتطبيق ما يمليه ذاك المرشد ويستجيبون على التغذية المرتجعة" .

عرض فريق مايكروسوفت Start.com

في الشركات الكبيرة تعتبر الإجراءات و اللقاءات أمور اعتيادية. تهدر الأشهر بحجة التخطيط لمميزات و مناقشة التفاصيل لتحقيق أهداف متفق عليها من قبل الجميع بشأن "أنسب" شيء للعميل.

قد يكون هذا هو المنهج الصحيح لتقديم برمجيات مغلفة تنقل عبر مسافات طويلة من خلال الأقراص الممغنطة أو غيرها، ولكن مع شبكة الإنترنت فنحن نملك ميزة رائعة. فقط أشحنه! دع المستخدم يخبرك عن البرمجية ، فإذا كان هناك خطأ قم بإصلاحه في نفس اليوم ثم أشحن البرمجية من جديد إلى الويب ! لا يوجد ما هو أصعب من مقاومة العميل لعقد الاجتماعات والنقاشات المطولة. فقط أشحن البرمجية لتثبت لهم أنك جاد ومثابر على تقديم وتطوير المنتج.

الأقوال أسهل بكثير من الأفعال – هذا يعني:

التخطيط لأشهر ليس ضرورياً.
ليس من الضروري أن تتفرغ لأشهر من أجل كتابة المواصفات. المواصفات يجب أن تكون أسس ثابتة و تفاصيل واضحة وتتم مراجعتها طيلة فترة التطوير. لا تحاول مناقشة كل التفاصيل لتغلق كافة القضايا المفتوحة قبل أن تشر ع في مرحلة التطوير.

اعرض ميزات أقل، لكن ذات نوعية ممتازة .
لا تتبع أسلوب الإطلاق الانفجاري أي أن تطرح إصدارا جديدة بمجموعة من المميزات. أعط المستخدمين مجموعات صغيرة الحجم حتى يتمكنوا من استيعابها.

إذا وجدت أخطاء بسيطة في التطبيق، لا تبالي قدم التطبيق بناء على السيناريو الذي حددته مسبقاً. بعدها قم بإصلاح تلك الأخطاء الموجودة في تطبيق الويب وبشكل تدريجي. كلما تمكنت من الحصول على التغذية المرتجعة بشكل أسرع كان ذلك أفضل. الأفكار على الورق تبدو مثالية ولكن بالتطبيق تتلاشى تلك المثالية. كلما أسرعت باكتشاف المشاكل الرئيسية التي تؤدي إلى فشل الفكرة ، كان ذلك أفضل.

التكرار السريع و التفاعل مع التغذية المرتجعة سيبني علاقة قوية بينك وبين العميل. تذكر أن الهدف هو كسب العميل من خلال تقديم ما يحتاجه.

—Sanaz Ahari، مدير برنامجStart.com, Microsoft

الفصل الثاني: خط الإنطلاق

انجاز القليل

أعمل أقل من منافسيك

الحكمة السائدة تقول لتنتصر على منافسيك لا بد أن تفوقهم في الإنفاق . فإذا كان لديهم أربع مميزات فأنت بحاجة إلى خمس أو 15 أو 25 . و إذا أنفقوا مبلغ من المال عليك أن تنفق ضعفه .و إذا كان لديهم 20 موظف فأنت بحاجة إلى 30 .

هذا الاندفاع هو سمة العقليات المتخلفة ،فهو حتماً سيقودنا إلى الهاوية . وهو كذلك مكلف و وسيلة دفاعية في بناء المنتجات لا تقوم بها إلا الشركات المترددة. الشركات المترددة و الدفاعية لا تفكر بالمستقبل بقدر ما هي تعيش في الماضي، وهي مقلدة لا رائدة .

إذا كنت تسعى لإنشاء شركة مقلدة ، عليك أن لا تكمل قراءة هذا الكتاب .

ماذا تفعل إذاً ؟ الإجابة في إنجاز القليل . أفعل أقل مما يفعله منافسوك لتنتصر عليهم . أبدأ بحل المشاكل البسيطة وأترك المشاكل الصعبة و المعقدة لأي شخص أخر . بدلاً من الاندفاع وتقديم كل شيء جرب أن تقدم القليل . بدلاً من انجاز كل شيء جرب أن تنجز القليل .

سنغطي مفهوم انجاز القليل من خلال هذا الكتاب ، لكن للمبتدئين فإن القليل يعني :

- أقل في المميزات .
- أقل في الخيارات أو التفضيلات .
- أقل في الأفراد وهيكل الشركة .
- أقل في الاجتماعات و الأفكار .
- أقل في الوعود .

ما المشكلة التي تواجهها ؟

اصنع البرمجيات لنفسك

يعتبر البدء بحل مشاكلك الشخصية طريقة رائعة لبناء البرمجيات.لأنك ستكون الجمهور المستهدف الذي يعرف ما المهم وما هو غير المهم. هذا الطريقة بداية عظيمة ستمنحك فرصة تقديم منتج ثوري .

المهم هنا أن تفهم أنك لست وحيداً .فإذا كانت لديك مشكلة فمن المحتمل أن مئات الآلاف غيرك لديهم نفس المشكلة . السوق سوقك، أليست هذه الطريقة سهلة ؟

ظهر تطبيقBasecamp من خلال مشكلة واجهناها: كشركة تصميم احتجنا إلى طريقة بسيطة لنتواصل مع عملائنا حول المشاريع . بدأنا ذلك من خلال شبكات خاصة (اكسترا نت) والتي كانت تحدّث يدوياً . لكن المشاريع التي تحتاج إلى تحديث دائم يعتبر تغيير الهتمل HTML يدوياً إجراء غير عملي . مواقع تلك المشاريع كانت دائماً غير ممتعة لدرجة أنه تم هجرها في نهاية المطاف. و كان هذا محبطاً لأنه جعلنا عشوائيين و العملاء في جهلهم يعمهون.

لذا بدأنا في البحث عن خيارات الأخرى .وجدنا أن كل أداة أما أنها 1) لا تقدم ما نريده أو أنها 2) متضخمة بمميزات لا نحتاجها ، مثل طرق السداد وطرق وصول صارمة و المخططات والرسومات الخ . عرفنا أنه لابد أن يكون هناك طريق أفضل ، لذا قررنا أن نبني برنامجنا الخاص .

عندما تحل مشكلتك الخاصة هذا يعني أنك أنشأت أداة منسجم معها. الانسجام والحب هو الأساس . الحب يعني أنك فعلاً ستستخدم أداتك و ستهتم بها . وهذا أفضل طريق لتجعل الآخرين يحبونها أيضاً .

حك جلدك بنفسك

يؤمن عالم المصادر المفتوحة منذ انطلاقته بتعويذة "حك جلدك بنفسك" . وهي تعني لمطوري المصادر المفتوحة حصولهم على ما يريدون من أدوات ومن ثم إعادة تشكيلها و تقديمها بالشكل الذي يريدونه. في الحقيقة الفائدة أكبر من ذلك بكثير .

كمصمم أو مطور لتطبيق ما ،في كل يوم ستواجهك مئات القرارات الصغيرة : أخضر أو أزرق ؟ جدول أو اثنان ؟ ساكن أو متحرك ؟ إلغاء العملية أو استعادتها؟ كيف يمكننا اتخاذ مثل هذه القرارات؟ قد نقوم بسؤال الآخرين إذا كانت أموراً تبدو لنا مهمة أما غير المهمة نقوم بعملية التخمين . تزداد هذه التخمينات لتصبح كالشبكة المترابطة من الافتراضات ، لتكون في الأخير عبء على تطبيقاتنا.

أكره هذا الشيء ، بصفتي مطور. معرفة كل التفاصيل الدقيقة في التطبيقات هو أمر مقلق. مطوري المصادر المفتوحة يقومون بأعمالهم بأنفسهم ولا يزعجهم ذلك. لأنهم هم أنفسهم المستخدمين ، و يعرفون الإجابة الصحيحة لـ 90% من القرارات التي يجب أن يتخذوها. أعتقد أن هذا أحد الأسباب التي تجعل الأشخاص يعودون لمنازلهم بعد يوم شاق من كتابة الأكواد ثم يقومون بالعمل على المصادر المفتوحة لأنها بالنسبة لهم نوع من الترويح عن النفس.

ــــديف توماس ، المبرمجون الواقعيون

وليدة الحاجة

شركة كامبيان مونيتور وُلدت من الحاجة . في السنوات الماضية كنا محبطين بسبب عدم جودة خيارات التسويق بالبريد الإلكتروني المتوفرة لنا . فمثلاً أحد الأدوات تقوم بعمل س و ص و لا تعمل ع والأخرى تقوم بعمل ص و ع لكن لا تستطيع عمل س . لم نرتح لهذا الشيء .

قررنا أن نلغي خطتنا لنبدأ بأنفسنا في بناء أداة تسويق بالبريد الإلكتروني .قررنا أن لا ننظر إلى ما فعله الآخرون لنبدأ في بناء أداة تناسبنا وعملائنا.

وفي نهاية المطاف تبين لنا أن هناك كثيرون غيرنا غير سعداء بالخيارات المتوفرة. لذا قمنا بعمل تعديلات طفيفة على البرنامج بحيث يمكن لأي شركة تصميم استخدامه . في أقل من ستة أشهر ، آلاف المصممين بدأوا يستخدمون كامبيان مونيتور في إرسال نشرات البريد الإلكتروني لأنفسهم ولعملائهم.

ــــديفيد قراينر ، مؤسس، كامبيان مونيتور

اعتن به

عندما تريد كتابة كتاب لا يكفي أن يكون لديك قصة مشوقة بل يجب أن يكون لديك رغبة في رواية القصة. ويجب أن يكون لشخصيتك حضور في تلك الكتابات. إذا كنت ستعيش مع شيء ما لمدة سنتين أو ثلاث أو بقية حياتك ، اعتن به .

—مالكوم قلاد ويل، مؤلف (المصدر : مقتطفات قليلة من مالكوم قلاد ويل)

مول نفسك

التمويل الخارجي هو الخطة البديلة

الأولوية لدى العديد من الشركات الناشئة هي الحصول على التمويل من المستثمرين. لكن تذكر أنك إذا اتجهت للتمويل الخارجي فأنه يلزمك الرضوخ لقوانين المستثمرين. والمخاطرة ستكون كبيرة .كذلك المستثمرين يرغبون باسترداد أموالهم وبسرعة. الحقيقة المرة أن تحقيق المكاسب المالية في الغالب يكون نقطة بدء انحدار جودة المنتج.

هذه الأيام لا تحتاج الكثير لتبدأ العمل. فالأجهزة رخيصة وهناك وفرة في برامج البنى التحتية المفتوحة المصدر والمجانية .جذب العملاء لم يعد يعتمد على السعر.

لكي تفعل ما تريده في المبلغ الذي معك، فكر ملياً وحدد ما الأمور الأساسية وما الذي تستطيع عمله في حالة عدم توفرها. ما الذي تستطيع إنجازه مع ثلاث أشخاص بدل من عشرة؟ ما الذي تستطيع إنجازه بـ 20 ألف دولار بدلاً من 100 ألف دولار ؟ ما الذي تستطيع إنجازه في ثلاث أشهر بدلاً من ستة ؟ ما الذي تستطيع إنجازه إذا احتفظت بوظيفتك وجعلت بناء برنامجك عمل إضافي ؟

العوائق تدفع بك إلى الإبداع

العمل ضمن موارد محدودة سيجعلك تخطط وبجدية تامة لتتعامل مع العوائق مبكراً، و هذا أمر جيد. العوائق تدفع بك إلى الابتكار.

شيء أخر جيد وهو أن العوائق تجبرك على إطلاق فكرتك إلى العالم عاجلاً غير آجل .و بعد شهر أو شهرين من العمل ستكون لديك فكرة جيدة جداً بشأن ما إذا كنت على المسار أم لا . إذا كنت على المسار الصحيح ستكون لديك بعد فترة قصير قدرة الاستمرار الذاتي ولن تحتاج إلى تمويل خارجي . وإذا كانت فكرتك عديمة الفائدة فعليك البدء من جديد مرة أخرى . على الأقل أنت ستعرف مدى جودة فكرتك الآن بدلاً من المستقبل أي بعد أشهر (أو سنوات) . و تستطيع أن تتراجع بسهولة . خطط الانسحاب تتعقد إذا كنت مرتبطاً بمستثمرين آخرين .

إذا قمت بعمل برمجيات فقط للحصول على المال سريعاً سيظهر ذلك . الحقيقة تقول أن الأرباح السريعة أمر مستبعد . لذا ركز على بناء أداة ذات جودة تستطيع أنت وعملائك العيش معها لفترة طويلة .

مساران

(جاك ووكر بدأ شركة (Disclive) مع ممول و بدأ شركة أخرى (The Show) بتمويل ذاتي . وهنا يناقش الاختلافات بين المسارين .)

أساس كل المشاكل لم يكن جمع المال نفسه ، لكن كل شيء أتى معه . طموحاتنا كانت كبيرة . يأخذ الأشخاص رواتب ، ليتم بعدها البدء في عملية البناء السريع ليتم في الأخير البيع أو البحث عن أي طرق أخرى، والدافع خلف ذلك هو استعادة أموال المؤسسين . في حالة الشركة الأولى ببساطة بدأنا نتظاهر بأننا أكبر بكثير مما كنا عليه – بدافع من الضرورة ...

مع The Show أدركنا أنه من الممكن تقديم منتج أفضل وبتكلفة أقل وكل ما نحتاجه هو المزيد من الوقت. نحن قامرنا بالقليل من أموالنا لأن الناس ستتقبل انتظار الجودة العالية على حساب السرعة . لكن الشركة بقيت (ويرجح أن تستمر كذلك) شركة صغيرة . و منذ بداية المشروع قمنا

بتمويل ذاتي كامل . و مع القليل من الشروط الإبداعية من قبل بائعينا ، لم نعد نحتاج لوضع المزيد من أموالنا في الشركة . ولم تعد طموحاتنا تقتصر على البناء والنمو و من ثم البيع ، إنما النمو من أجل النمو كي تستمر الاستفادة منها مالياً .

—تعليق من Signal vs. Noise

ثبت الميزانية والتوقيت.. و اجعل الخيارات مرنة

الإطلاق في حدود الوقت والميزانية

هناك طريقة سهلة للإطلاق في حدود الوقت والميزانية المخطط لها وهي :اجلعهما ثابتين ، فلا تنفق المزيد من الوقت أو المال في مشكلة ما ،فقط قلص نطاق خيارات ذلك الشيء .

هناك أسطورة تقول : نستطيع أن نطلق منتجنا بكافة الخيارات التي تريدها في الوقت والميزانية المحددة. هذا الشيء مستحيل ولو حدث فإنه حتماً سيكون على حساب الجودة .

إذ لم تستطيع إنجاز كل شيء في الوقت والميزانية المحددة ، رجاءً لا تقم بزيادتهما . بدلاً من ذلك قلص نطاق الخيارات . دائماً ما يكون هناك وقت لإضافة الأشياء في وقت لاحق– القادم باقي أما الحاضر فهو ذاهب لا محالة .

إطلاق شيء رائع بخيارات أقل مما خطط له أفضل من إطلاق شيء متوسط المستوى مليء بالأخطاء بحجة الرغبة في تنفيذ كل الخيارات في وقت وميزانية محدودة جداً. دع السحر للسحرة . فأنت بدأت في العمل الحقيقي من أجل تشغيله وإيصال المنتج.

وهذه فوائد ثبات الوقت والميزانية و مرونة نطاق الخيارات :

- **ترتيب الأولويات**

يجب أن تعرف ما المهم فعلاً . ما الذي ستقدمه في الإصدار الأولي ؟ فهذا الشيء يزيد الضغط عليك مما يدفعك لاتخاذ القرارات الصعبة بدلاً من التردد فيها.

- **الواقعية**

تحديد الطموحات أمراً أساسي . إذا كنت تحاول وضع وقت وميزانية ونطاق ثابت فلن يكون بمقدورك تقديم مستوى عال من الجودة . بالتأكيد يمكنك تقديم شيء ما ، لكن "شيئاً " هل هي ما تريد تقديمه حقاً ؟

- **المرونة**

القدرة على التغيير أمراً أساسي. وضع كل شيء ثابت يجعل من التغيير أمراً عسيراً . النطاق المرن سيجعلك تقدم خيارات بناء على تجربتك الحقيقية في بناء ذلك المنتج . المرونة هي صديقك العزيز .

توصيتنا : قلل من نطاق الخيارات . من الأفضل أن تقدم نصف المنتج على أن تقدم منتجاً نصف غبي (المزيد حول هذا لاحقاً) .

واحد ، اثنين ، ثلاثة ...

كيف يمكن لمشروع أن يتأخر سنة كاملة عن موعده ؟ التأخر يوم واحد في كل مرة .

—فريد بروكس، عالم حاسوب ومهندس برمجيات

ليكن لديك عدو

أبدأ بالقتال

في بعض الأحيان يكون أفضل طريق لمعرفة ما الذي يجب أن يكون عليه تطبيقك هو معرفة ما لا يجب أن يكون عليه . تعرف على أعداء تطبيقك لتعرف أين تكون وجهتك.

عندما قررنا إنشاء برنامج إدارة مشاريع ، كنا نعرف أن تطبيق مايكروسوفت Microsoft Project مسيطر ولكن بدلاً من الخوف منه استخدمناه كمحفز . لذا قررنا أن يكون Basecamp تطبيقا ً مختلفا ً بشكل كامل ومنافساً عنيداً لتطبيق مايكروسفت.

أدركنا أن إدارة المشاريع ليست مخططات و رسوم وتقارير و إحصائيات ، بل هي نظام اتصالات بين العاملين ، كما أن نجاح المشروع لا يرتبط بمدير المشروع فقط بل مسؤولية كل فرد في تلك المنظومة .

عدونا كان إدارة المشاريع الدكتاتورية وما تستخدمه من أدوات لتعقيد العمل . أردنا إضفاء الطابع الديمقراطي على إدارة المشاريع ، بمعنى جعل كل شخص جزء منها (بما في ذلك العملاء) . فالمشاريع يتم إنجازها بشكل أفضل عندما يكون هناك مسؤولية جماعية يتحملها كل شخص .

فيما يتعلق بخدمة الرايت بورد Writeboard كنا نعرف بوجود منافسين يملكون الكثير من المميزات الناجحة. لذا قررنا أن نركز على مبدأ "اللا فوضى" في التطبيق . أنشأنا تطبيقاً يسمح للأشخاص بتبادل الأفكار والتعاون فيما بينهم ببساطة دون إرباكهم بمميزات غير ضرورية، فكل ما هو غير ضروري تم استبعاده. وبعد ثلاث أشهر فقط من الإطلاق كان هناك أكثر من 100,000 ملف تم إنشاؤه في خدمة الرايت بورد.

عندما بدأنا في تطبيق Backpack كان عدونا الهيكليات و القواعد الصارمة . فالناس يجب أن يقوموا بتنظيم معلوماتهم بطريقتهم الخاصة وليس بناء على سلسلة من الشاشات المهيئة مسبقا ً أو عدد كبير من حقول النماذج المطلوبة .

أهم ايجابيات وجود منافس هو أنه يكون لك رسالة تسويقية واضحة المعالم . الناس في هذه الحالة يعيشون حالة الصراع ، لذا فهم سيفهمون المنتج بمقارنته مع المنتجات الأخرى . مع العدو المحدد مسبقا ً ستتمكن من تقديم كل ما يرغبه الآخرين. لن يكتفي الآخرين بفهم المنتج وحسب بل ستتفاجأ أنهم سيدعمونه. أن هذا طريق مضمون لشد الانتباه و تأجيج العاطفة .

ومع كل ما قيل سابقا ً، من المهم أيضا ً أن لا تصل المنافسة إلى حد الهوس. الإفراط في تحليل المنتجات الأخرى سيحد من تفكيرك. حاول أن تلقي نظرة على من حولك ثم تقدم لتحقق رؤيتك و أفكارك.

لا تتبع القائد

المسوقون بل الناس بشكل عام مدربون جيدا ً على إتباع القائد . فالغريزة البشرية دائما ً ما تحاول التعرف على ما يحدث في المنافسة ثم تحاول التفوق عليه ، كأن تكون أرخص من المنافس الذي يتنافس معك على السعر أو أسرع من المنافس الذي يتنافس معك على السرعة .المشكلة في هذا أنه بمجرد أن يقوم مستهلك ما بالشراء سيرى مستهلك أخر الكذبة ، إقناع المستهلك بالتحويل هو نفسه إقناعه على قبول أنه كان على خطأ . والناس يكرهون الإقرار بأنهم كانوا على خطأ .

بدلا ً من ذلك ، أخبر عملائك بقصة مختلفة ، وأقنعهم بقصتك و أنها أهم من القصة التي يؤمنون بها حاليا ً . إذا كان منافسيك أسرع يجب أن تكون أرخص . إذا كانوا يقدمون قصة صحية قدم قصة مقنعة . العملية ليست تحديد س و ص على المحاور للإدعاء "بأننا الأرخص" ، بقدر ما هي الإخبار بقصة حقيقية تختلف تماما ً عن القصة السائدة.

—سيث قودن ،مؤلف/مبادر (المصدر كن كذابا ً بشكل أفضل)

ما المشكلة الرئيسية ؟

النظر إلى ما يعمله المنافسون هو أسرع طريقة تؤدي بك إلى المشاكل. هذا الشيء ينطبق علينا خصوصاً في تطبيق BlinkList . منذ إطلاقنا له كان هناك 10 خدمات متخصصة في مشاركة الروابط مع الآخرين . بعض الأشخاص بدؤوا في عمل جداول مباشرة على الإنترنت مع تفاصيل مقارنة ميزة بميزة لتلك الخدمات.

من ناحية ثانية ،هذا الشيء يظلل الشخص سريعاً . بدلاً من ذلك ، بقينا نركز على الصورة العامة ونسأل أنفسنا ما المشكلة الأساسية التي نحاول حلها و كيف يمكننا حلها .

—مايكل رينوق ، أحد مؤسسي ، MindValley و Blinklist

لا تجعل عملك رتيباً

مدى حبك لتطبيقك من عدمه سيبدو واضحاً للجميع

كلما ابتعدت عن النظر إلى عملية بناء التطبيق بأنها عمل رتيب ، كلما كان عملك أجود. أجعل برنامجك صغيراً وقابلاً للإدارة لكي تستمتع فعلاً بالعملية .

إذ لم تشعر بالإثارة وأنت تتعامل مع برنامجك فهناك خطأ ما ، و إذا كنت تعمل من أجل تحقيق ربح سريع فسترى ذلك. وبالمثل إذا كنت محباً ومتعاطفاً مع تطبيقك فسيبدو هذا جلياً في المنتج النهائي . يملك الناس القدرة على قراءة ما بين السطور.

حضور العاطفة والمشاعر

في التصميم ، حيث غالباً ما يكون المعنى غامضاً وبشكلٍ قاتل، قليلة هي الأشياء التي تكون أكثر وضوحاً و إدراكاً من العاطفة والأحاسيس. وهذا صحيح سواء في تصميم منتج يسعدك أو لا يؤثر فيك ، ففي كلا الحالتين من السهل أن نلاحظ اللمسات العاطفية للسواعد التي قامت ببنائه.

بطبيعة الحال الحماس سيظهر نفسه بنفسه ، وبالمثل عدم المبالاة لن تُنسى. إذا كان التزامك لا يهتم بالعاطفة في عملاً ما، بغض النظر عن مدى جاذبية وإتقان تصميمه ،سيصبح هذا العمل خاوياً ومن المستحيل إخفاء هذا الشيء.

—كوي فن ، Subtraction.com

المخبز

التجارة الأمريكية في هذا المجال تتمحور في الحقيقة حول تطوير الفكرة و جعلها مربحه و بيعها مادامت مربحه ومن ثم الانسحاب أو التنويع. هي ببساطة امتصاص لكل شيء. فكرتي كانت الاستمتاع بالخَبزَ و بيع الخبز الخاص بي ، الناس سيحبونه وبالتالي بيع المزيد . سيستمر المخبز لأنك تقدم طعاماً جيداً والناس سعداء بذلك .

—إيان ماكي ، عضو في الفيوقازي و مشارك في ملكية دسكوردز ريكوردز
(المصدر الأشخاص في Salon.com | إيان ماكي)

الفصل الثالث: كنّ موجز

أقل حجماً

أكثر إيجازاً، أسهل تغييراً

إذا الشيء ازداد ضخامةً، تطلب طاقة أكبر لتغييره. هذا صحيح في عالم الأعمال، وكذلك نرى في المحيط من حولنا.

عندما نأتي لتطبيقات الويب، التغيير لابد أن يكون يسير وبأقل تكلفة ممكنة. إذا كنت لا تستطيع التغيير بشكل سريع، فستخسر الساحة لأجل شخص آخر. لهذا السبب عليك أن تركز على أبسط صورة ممكنة لبناء التطبيق.

الحجم يزداد بواسطة ...

- عقود على المدى الطويل
- فائض في عدد الموظفين
- قرارات دائمة
- اجتماعات حول اجتماعات أخرى
- عملية كبيرة
- قوائم (مادية أو معنوية)
- الأجهزة، البرامج والتكنولوجيا المستوردة
- نماذج البيانات الخاصة
- الماضي يحكم المستقبل
- خرائط طويلة المدى
- قوانين المكتب

الحجم ينخفض بواسطة ...

- التفكير في الوقت المناسب
- فريق أعضاؤه متعدد المهام
- الاهتمام بالقيود، و عدم محاولة تركها
- برامج أقل، كود أقل
- ميزات أقل
- فريق صغير الحجم
- البساطة
- واجهات سهلة وغير معقدة
- منتجات مفتوحة المصدر
- صيّغ البيانات المفتوحة
- ثقافة مفتوحة تجعل من السهل الاعتراف بالأخطاء

حجم أقل يتيح لك إمكانية تغير الاتجاه بسرعة، يمكنك أن تتفاعل و تتطور ويمكنك أن تركز على الأفكار الجيدة وتقذف السيئة منها ، يمكنك أن تسمع وتستجيب لزبائنك، ويمكنك أن تدمج التقنيات الجديدة حالياً بدلاً من تأخير ذلك لوقت لاحق. وعوضاً عن امتلاك حاملة طائرات لما لا تمتلك قارباً يمكنك توجيهه وأنت تدخن سيجارتك أو تحتسي شرابك ؟

فعلى سبيل المثال، لنتصور بإيجاز ؛ شركة ذات حجم أقل بنّت منتج أقل برمجياً وأقل في الميزات. على الجانب الأخر شركة بحجم أكبر حصلت على هذا المنتج مع إضافة برمجيه كبيرة وميزات أكثر. ثم لنفترض انتشار تكنولوجيا جديدة كالأجاكس أو مفهوم جديد كـ الأوسمة في جميع الأرجاء . من سيكون قادراً على تكييف منتجاته بشكل أسرع؟ فريق مع برامج أكثر وميزات أكثر و خريطة عمل لمدة 12 شهراً أو فريق ببرامج أقل وميزات أقل و أكثر تألفاً. لما لا نُفّعل مرحلة " التركيز على ما نحن بحاجة لـ تسليط الجهد عليه حالياً " ؟ .

من الواضح أن الشركة ذات الحجم الأقل تكون في وضع أفضل للتكيف مع المطالب الفعلية للسوق. ومن المرجح أن الشركة ذات الحجم الأكبر لا تزال تناقش التغييرات أو تدفعها خلال عملية بيروقراطية طويلة وبعد أن قامت الشركة ذات الحجم الأقل بالتغيير. شركة الحجم الأقل ستمضي

خطوتين للأمام في حين شركة الحجم الأكبر لازالت تخطط كيف تمشي.

بذكاء ومرونة، أعمال الحجم الأقل تستطيع بسرعة تغيير كامل نموذج العمل، المنتج، سمة المجموعة و رسالة التسويق. من الممكن أن يعملوا أخطاء ويصلحوها بسرعة. من الممكن تغيير أولوياتها، تشكيلة المنتجات، والرؤية. والأكثر أهمية، **يمكنهم تغيير أذهانهم**.

اخفض كلفة التغيير

ابق مرناً عن طريق حد العقبات التي تحول دون التغيير

التغيير هو أفضل صديق، والأكثر تكلفة هو إجراء تغيير، الأقل هو أفضل ما تفعله. وإذا كان منافسوك يستطيعون التغيير أسرع منك، فانك تمتلك نقيصة كبيرة. إذا التغيير ينتج تكلفة كبيرة، فأنت ميت.

هنا حيث البقاء صغيراً يساعدك حقاً و القدرة على التغيير الدائم يكون شيء مميز يملكه الفريق الصغير تلقائياً**!** الأمر الذي لا يمكن انه تملكه الفرق الكبيرة أبداً. هنا حيث يحسد الرجال الكبار رجالاً صغار. الأسابيع التي يمكن أن يأخذها فريق كبير في منظمة ضخمة للتغيير من الممكن أن تستغرق يوماً في منظمة صغيرة وضعيفة. هذه الميزة التي لا تُقدر بثمن، رخص وسرعة التغييرات هي السلاح السري للأصغر.

وتذّكر: جميع النقد، جميع التسويق، وجميع الناس في العالم لا يمكن أن تشتري لك خفة الحركة التي تحصل عليها من البقاء صغيراً.

عندما يتعلق الأمر بتكنولوجيا الويب، التغيير يجب أن يكون سهل ورخيص. إذا لم تستطع التغيير بسرعة الطيران، سوف تخسر انطباع شخص ما قد يكون ذا أهمية، لهذا السبب أنت تحتاج أن تتمسك بالحجم الأقل.

النشوء

النشوء هو واحد من المبادئ الأساسية لخفة الحركة وهو الأقرب للسحر النقي. خصائص النشوء ليست مُصممة أو داخلية البناء، هي تحصل ببساطة كنتيجة ديناميكية لراحة النظام. "النشوء" أتت من منتصف القرن السابع عشر بمعنى "غير متوقعة الحدوث". لا يمكن التخطيط لها أو جدولتها ولكن يمكن زراعة بيئة تمكنك من تحقيقها والاستفادة منها.

المثال الكلاسيكي للـ النشوء تكمن في تجميع سلوك الطيور، المحاكاة بالكمبيوتر يمكن أن تستخدم ثلاث قواعد بسيطة (مثل "لا تضعها بجانب بعضها") وفجأة ستحصل على سلوك معقد جداً كتجميع القطيع. لا شيء من هذا السلوك المتقدم (مثل إصلاح نفس الشكل مرة أخرى) يمكن تحديدها بقواعد أو خطوات، إنها تنشئ تلقائياً من القوى المحركة للنظام.

قواعد بسيطة، كما هو الحال مع محاكاة الطيور، تؤدي إلى سلوك معقد. قواعد معقدة، كما هو الحال مع قانون الضرائب في معظم البلدان، تؤدي إلى سلوك غبي.

Mالعديد من الممارسات الشائعة لتطوير البرامج لديها آثار جانبية مؤسفة للقضاء على أي فرصه في سلوك النشوء. معظم محاولات التحسين — تحاول تخفيض شيء ما بكل وضوح — وتقليل اتساع ومدى التفاعلات والعلاقات، التي هي مصدر النشوء. في مثال تدفق الطيور، على نحو تصميم جيد للنظام، تكون التفاعلات والعلاقات التي تنشئ السلوك المثير للاهتمام.

الأصعب إننا نشدد الأمور للأسفل، هناك مجال للإبداع و للحلول الناشئة وسواء كان ذلك بقفل المتطلبات للأسفل قبل أن تُفهم جيداً أو تحسين الكود قبل الأوان أو اختراع ملاحة معقدة و تدفق سينورهات العمل قبل السماح للمستخدمين النهائيين باللعب مع النظام؛ فان النتيجة واحدة: إفراط في التعقيد، نظام غبي بدلاً من وضوح وأناقة نظام يسيطر عليه النشوء .

أبقه صغيراً، أبقه بسيطاً، دعه يحدث.

—أندرو هنت (Andrew Hunt،)، المبرمجون الواقعيون

الفرسان الثلاثة

استخدم فريق من ثلاثة للإصدار 1.0

لأول إصدار من تطبيقك، ابدأ بثلاثة أشخاص. هذا هو العدد السحري الذي سيعطيك ما يكفي من القوى العاملة التي تسمح لك بالبقاء متيقظاً و محدثاً. ابدأ بـ مطور، مصمم و مهني (شخص يمكنه التجول بين كلا العالمين).

من المؤكد الآن انه يعد تحدياً لبناء تطبيق مع عدد قليل من الأشخاص، ولكن إذا حصلت على الفريق الصحيح فهو يستحق التحدي. الأشخاص ذوي الكفاءة والمهارة ليسوا في حاجة موارد لا نهاية لها، إنهم ينجحون في التحدي المتمثل في العمل على إطار القيود واستخدام قدراتهم الإبداعية لحل المشكلات. إن نقص القوى العاملة يعني انك ستضطر للتعامل مع التجارة في وقت مبكر من العملية — وهذا ممتاز. حيث ستجعلك تُشكل أولوياتك مبكراً وليس أجلاً. وستتمكن من التواصل دون القلق المستمر حول ترك الأشخاص خارج الحلقة.

إذا لم تتمكن من بناء نسختك الأولى بـ ثلاثة أشخاص، فأنت بحاجة إلى أشخاص مختلفين أو بحاجة إلى تقليل متطلبات نسختك الأولية. تذكر، من الجيد إبقاء نسختك الأولى صغيرة و مُحكمة. ستحصل بسرعة على رؤية حول إذا كانت فكرتك امتلكت أجنحة ، وإذا فعلت ذلك، ستكون قاعدة واضحة وبسيطة للبناء عليها.

قانون Metcalfe و فُرق المشروع

ابقْ الفريق صغيراً قدر الإمكان، يقول قانون Metcalfe " إن قيمة الاتصالات في أي نظام تنمو تقريبياً بتربيع عدد مستخدمي النظام" وتكون نتيجة طبيعية عندما يتعلق الأمر بفريق المشروع: كفاءة الفريق تقريباً عكسية عند تربيع عدد الأعضاء في الفريق . أنا اعتقد أن البداية بثلاثة أشخاص هي الأمثل لإطلاق التطبيق 1.0 . ابدأ بخفض عدد الأشخاص الذين تخطط لإضافتهم للفريق ثم خفّض عدد أكثر.

—مارك هدلند (Marc Hedlund)، منظم مقيم في وسائل الإعلام، O'Reilly Media

تدفق الاتصالات

تنساب الاتصالات بسهولة أكثر في الفرق الصغيرة أكثر من الفرق الكبيرة، إذا كنت الشخص الوحيد في المشروع؛ فالاتصال سهل. مسار الاتصال الوحيد هو بينك وبين العميل. وكلما زاد عدد الأشخاص في المشروع، كذلك يكون عدد مسارات الاتصالات. كلما زاد عدد الأشخاص، ستتضاعف الاتصالات، وتتناسب مع تربيع عدد الأشخاص.

—ستيف ماكونيل (Steve McConnell) ، في Construx بُناة البرمجيات المحدودة
(من الأقل هو الأكثر: قفزات البدء الانتاجية مع الفرق الصغيرة.)

تعايش مع القيود

دعْ القيود ترشدك للحلول المبدعة.

لا يوجد ما يكفي للاستمرار بالسير، الوقت لا يكفي، المال لا يكفي، الأشخاص لا يكفون.

هذا شيء جيد.

بدلا من الضيق ذرعاً بهذه القيود، اكسرها ودعها ترشدك. القيود تسوق روح الابتكار و قوة التركيز. وبدلاً من محاولة إزالتها استخدمها كميزة خاصة بك.

عندما بنى فريق 37signals مشروع Basecamp، كان لدينا العديد من القيود، لدينا:

- تصميم الشركة لتعمل
- عمل قائمة بالعملاء
- فارق التوقيت بسبع ساعات (ديفيد كان يبرمج في الدنمارك ، البقية منا في الولايات)
- الفريق صغير
- لا يوجد تمويل خارجي

شعرنا بـ مشكلة "لا يكفي" لذا حافظنا على مشروعنا صغيراً. بهذه الطريقة فقط استطعنا وضع الكثير عليه. تولينا مهام كبيرة و قسمناها إلى أجزاء صغيرة تمكننا من معالجتها في وقت واحد. لقد تحركنا خطوة بخطوة و رتبنا أولوياتها كما أردنا طويلاً.

هذا أجبرنا على الخروج بحلول مبدعة وخلاقة، خفضنا كلفة التغيير باستمرار بناء برمجيات اقل و أعطينا العملاء خصائص كافية لحل مشكلاتهم بطريقتهم الخاصة، فارق التوقيت والمسافة بيننا جعلتنا أكثر كفاءة و فعّالية في تواصلنا وبدلاً من الاجتماع بشخص، تواصلنا حصرياً بطريقتين: التراسل الفوري (IM) والبريد الالكتروني الأمر الذي دفعنا للوصول بسرعة للهدف.

القيود تكون غالباً مميزات في صورة مموهة؛ أنسى أمر رأس المال الاستثماري، دورات إطلاق الإصدارات الطويلة والتوظيف السريع و بدلاً عنها: اعمل بما لديك.

كافح الفساد

ما وُصف بأنه "موضة الاختلاس" من الأفضل وصفه بـ "ملامح الفساد" مثل الفطريات على الكوكب، إنها تنتشر تدريجياً وتطمس حقيقة سمات المنتج في حين أنها تستنزف العملاء الساذجين. الدواء المضاد لملامح الفساد هو "صفقة محدودة بهدف" حيث التلاعب في الخصائص يكون ملغي بالنسبة إلى الوقت الذي ستستغرقه الصفقة لتنفيذها. يكون هذا غالباً في حالة الخصائص الأكثر نفعاً التي تأخذ ملامح أطول لتنفيذها، وهكذا مزيج من مكافحة الفساد بغرض إنتاج برمجيات كما نعلم وكما نحب.

—جف راسكن (Jef Raskin) ، كاتب (من "لماذا البرمجيات بالطريقة التي هي عليها")

كنْ كما أنت

ميّز نفسك عن أكبر الشركات بكونك صديق و إنسان

الكثير من الشركات الكبيرة تعمل خطأ بمحاولتها التمثيل أنها عملاقة، إنها كمن تصور حجمها بهزال وضعف يحتاج إلى تغطية، سيء للغاية. تستطيع أن تكون ميزة كبيرة لكونها صغيرة، خاصة عندما يتعلق الأمر بالاتصالات.

تتمتع الشركات الصغيرة بشكليات اقل، بيروقراطية أقل، وحرية أكثر. **بشكل افتراضي؛ الشركات الصغيرة هي الأقرب إلى العملاء.** مما يعني أن بإمكانهم التواصل مباشرة وبطريقة أكثر ألفة مع العملاء. إذا كنت صغيراً؛ يمكنك استخدام لغة مألوفة بدلاً من الرطانة. موقعك ومنتجك يمكن أن تمتلك صوت بشري بدلاً من صوت الشركات كالرجل الآلي. كونك صغيراً يعني انك تستطيع الحديث مع عملاءك وليس الانحناء لهم.

يوجد أيضاً ميزة الاتصالات الداخلية في الشركات الصغيرة جداً حيث يمكنك دفن الشكليات، لا حاجة لعمليات شاقة وتواقيع متعددة لموازنة كل شيء. الجميع في العملية يمكن أن يتكلم بصراحة وأمانة. تدفق الأفكار الغير مقيد هو واحد من مزايا البقاء صغيراً.

كن فخوراً، متحدٍ بصدق

قد تُفكر أن العميل قد ينخدع بالزيادات في عدد موظفين شركتك أو اتساع العروض التي تُقدمها، الأذكياء منهم، وهم العملاء اللذين تريدهم حقاً، سيعرفون دائماً الحقيقة — سواء عن طريق الحدس أو الخصم الذي تقدمه. بإحراج، كنت جزء من كذبة بيضاء في الماضي ولا أجد أن أي من

هذه الحالات قد نجحت في معظم المسائل التجارية: كـ الود، الاستمرار و علاقات المنفعة المتبادلة مع الأشخاص اللذين يمتلكون حاجة حقيقية لخدماتك المقدمة. الأفضل أن تكون فخوراً، متحدٍ بصدق حول الحجم الفعلي والاتساع للشركة.

—كوي فن (Khoi Vinh)، Subtraction.com

في أي وقت للجميع

لا يهم في أي الأعمال أنت، خدمة العملاء الجيدة هي المطلب الأكبر لأي عميل و يجب أن يحصل عليها. نحن نطالب بها لأي خدمة نستخدمها؛ إذن لماذا نعتقد أن عملاؤنا سيكونون في موضع مختلف؟ منذ البداية المبكرة جعلناها سهلة وشفافة للحصول على تواصل معنا لأي عدد من الأسئلة قد تكون لديهم، و على موقعنا أدرجنا قائمة بالأرقام المجانية المحولة إلى هواتفنا المحمولة و على بطاقة أعمالنا أدرج كلٍ منا قائمة هواتفه. كما نؤكد لعملائنا بإمكانية التواصل معنا في أي وقت، بغض النظر عن ما قد تكون المشكلة. يُقدر عملاؤنا هذا المستوى من الثقة ولم يسبق لأحد الحصول على هذه الخدمة.

—إدورد كنتل (Edward Knittel)، مدير المبيعات والتسويقKennelSource

---

الفصل الرابع: المهم فالأهم

ماهي الفكرة الرئيسية

بشكل واضح وجلي عرّف رؤية محددة لبرنامجك: ماذا يُساند؟ عن ماذا يكون حقيقة؟

قبل أن تبدأ في التصميم او كتابة الشفرات البرمجية، تحتاج الى معرفة الغرض من برنامجك - وهو ما يُعبر عنه بـ الرؤية . فكّر بعمق. لماذا يجب أن يُوجد هذا البرنامج؟ مالذي يجعله مختلفاً عن البرامج الأخرى المشابهة؟ .

هذه الرؤية ستُوجه قراراتك وتبقيك على طريق ثابت. حينما تجد نقطة اختلاف، إسأل: " هل نحن صادقون مع هذه الرؤية؟ "

أيضاً رؤيتك يجب أن تكون قصيرة، بحيث أن الجملة يجب أن تكون كافية لتعبر عن الفكرة كاملةً. في التالي رؤيتنا لكل منتجاتنا:

- **Basecamp**: إدارة المشروع تكمن في الاتصال .
- **Backpack**: إعادة نهايات الحياة الفاشلة سوية .
- **Campfire**: مجموعة دردشة خلال منصات IM .
- **Ta-da List**: منافسة من خلال تدوين الملاحظات .
- **Writeboard**: لوحة الكلمات .

في Basecamp، على سبيل المثال، كانت الرؤية " إدارة المشروع تكمن في الاتصال ." شعرنا بقوة أن الاتصال الفعال في المشروع يؤدّي إلى الملكية الجماعية، التدخّل، الاستثمار والتعاون . يحصل كل شخص على نفس الصفحة ويعملون نحو هدف مشترك. عرفنا إذا Basecamp أمكنه عمل ذلك؛ كل شئ آخر يسقط في الخط .

هذه الرؤية قادتنا لإبقاء Basecamp منفتح وشفاف بقدر الإمكان. وبدلاً من تحديد الاتصال من خلال الشركة، أعطينا عملائنا إمكانية الدخول أيضاً. اعتقادنا حول الرُخص والإذن يكون اقل ويزيد حول تشجيع كل المشاركين للاشتراك. إن الرؤية هي التي تغيّبنا عن المخططات، الرسوم البيانية، الجداول، التقارير، الإحصائيات وبرامج الجدولة وبدلاً من ذلك تركّز على أولويات الاتصال مثل الرسائل، التعليقات، عمل القوائم و مشاركة الملفات . اتخّذ القرار الكبير حول رؤيتك مقدماً وكل قراراتك الصغيرة المستقبلية تصبح أكثر سهولة.

فلسفة اللوحة البيضاء (Whiteboard)

أندي هنت (Andy Hunt) و أنا كتبنا ذات مرة تحويل لصفقة بطاقة المدين (debit card). كان المتطلب الرئيسي أن المستخدم لبطاقة المدين يجب أن لا يكون عنده نفس الصفقة طُبقت على حسابه مرتين. بصيغة أخرى لا يهم أسلوب نوع الفشل الذي قد يحدث، الخطأ يجب أن يجانب عدم معالجة الصفقة بدلاً من معالجة صفقة مضاعفة.

لذا، كتبناه بحروف كبيرة على لوحتنا البيضاء المشتركة : خطأ في مصلحة المستخدمين.

انضم إليه حوالي نصف درزن من حكم أخرى . سوياً؛ هذه الحكم قادت كل القرارات الصعبة التي اتخذناها خلال بناء شئ معقد. و معاً؛ أعطتْ هذه القوانين برنامجنا التحام داخلي قوي و ثبات خارجي رائع.

—ديف توماس (Dave Thomas) ، المبرمجون الواقعيون

اصنع انشودتك

تحتاج المنظمات علامات للطرق. انعم يحتاجون خلاصة؛ كل يوم عندما يستيقظ الموظفون يحتاجون معرفة لماذا هم ذاهبون للعمل. يجب أن تكون هذه الخلاصة قصيرة وحلوة، وتشمل : لماذا أنت موجود؟ ما لذي يحفزك؟ أنا أدعو هذا بالنشيد — ثلاثة أو أربعة كلمات تصف لماذا أنت هنا .

—قاي كاويسكي (Guy Kawasaki)، كاتب (من اصنع انشودتك)

تجاهل التفاصيل منذ البداية

اعمل من الكبير إلى الصغير

نحن مصابون بالهوس حول التفاصيل.

- المسافة بين العناصر.
- النوع المثالي من الإدارة.
- اللون المثالي.
- الكلمات المثالية.
- أربعة اسطر من الكود بدلا من سبعة.
- 90% مقابل 89% .
- 760 بكسل مقابل 750 بكسل .
- 39$/الشهر مقابل 49$/الشهر .

النجاح والرضا في التفاصيل .

على أية حال، النجاح ليس الشئ الوحيد الذي تجدُه في التفاصيل. ستجد أيضاً ركود، خلافات، اجتماعات، وتأخيرات. هذه الأشياء بإمكانها قتل الروح المعنوية وتقليل فرصَك من النجاحِ.

كمْ مرّة وجدت نفسك التصقت على تصميم مفرد أو عنصر في كود ليوم كامل؟ كمْ مرّة أدركت بان التقدم الذي أحدثته اليوم لم يكن تقدم حقيقي؟ هذا يحدث عندما تُركّز على التفاصيل مبكراً في العملية، سيكون هناك الكثير من الوقت الذي يوصلك إلى الكمال، فقط اعمله لاحقاً .

لا تقلق حول حجم العنوان البارز في الأسبوع الأول، أنت لست بحاجة لان تتسمر لإتقان ظل اللون الأخضر في الأسبوع الثاني، ولست بحاجة لتحريك زر "إرسال" ثلاثة بكسل إلى اليمين في الأسبوع الثالث. الآن ضع المادة في الصفحة فقط ثم استعملها و تأكّد من أنها تعمل ، ويمكنك لاحقاً من تعديلها و إتقانها.

تكشف التفاصيل نفسها عندما تستعمل ما أنت تبني، أنت سترى ما يحتاج إلى انتباه أكثر، أنت ستشعر ما هو المفقود، أنت ستعرف أين المطبات لتمهيدها لأنك ستستمر باستعمالها. وذلك عندما تحتاج لجذب انتباهك وليس قبل ذلك.

الشيطان في التفاصيل

تَغلبتُ حقاً على موقف " ادخل التفاصيل مباشرةً " بعد أن أخذت بعض الدروس في الرسم ... إذا بدأت في رسم التفاصيل مباشرة فأنت متأكد بان رسمك سيتناسق. في الحقيقة؛ أنت تتغيب عن النقطة الرئيسية بالكامل.

يجب أن تبدأ بفهم أبعادك بشكل صحيح للمشهد كاملاً، ثم ترسم اكبر الأجسام في مشهدك حتى تصل إلى اصغر واحد. الرسم يجب أن يكون حراً جداً حتى هذه النقطة، ثم تستطيع الاستمرار بتضليل حجم الأجسام التي تجلب الحياة إلى المشهد. تبدأ بثلاث درجات فقط (فاتح ، وسط ، غامق) يعطيك هذا رسم متناغم. ثم لكل جزء في رسمك تُعيد تقييم الثلاثة الظلال المتدرجة وتطبّقهم. اعملها حتى ترضى عن أحجام الأجسام (يتطلب تكراراً متعدداً)...

اعمل من الكبير إلى الصغير، دائماً .

—باتريك لايفليور (Patrick Lafleur)، إنشاء الأجسام المحدودة. (من الإشارة مقابل الضوضاء)

هي مشكلة عندما تكون مشكلة

لا تضييع الوقت على مشاكل لا تمتلكها بعد

هل أنت مضطر للقلق حول قياس 100.00 مستخدم يومياً عندما تحتاج سنتان للتحصل على هذا العدد؟

هل يتعين عليك استأجر ثمانية مبرمجين إذا أنت تحتاج إلى ثلاثة فقط اليوم ؟

هل تحتاج 12 خادم (servers) إذا استطعت تشغيل اثنان فقط لمدة سنة ؟

كن كالشراع

يقضى الناس غالباً وقتاً أكثر من اللازم يحاولون حل مشاكل مقدمة و التي لم تحصل بعد! لا تفعل ذلك. نحن أطلقنا Basecamp بدون القدرة على محاسبة العملاء! منذ أن حاسبنا المنتج في الدورات الشهرية، عرفنا بان لدينا فجوة 30-يوماً لحساب القيمة. استعملنا ذلك الوقت لحل مشاكل أكثر استعجالاً وبعد ذلك، بعد الانطلاق، عالجنا المحاسبة. لقد عمل جيداً (وأرغمنا على حل بسيط بدون الأجراس والصافرات الغير ضرورية).

لا تجهد المواد حتى تضطر لذلك. لا تبنْ زيادة . زدْ من الأجهزة وبرامج النظام حسب الضرورة. إذا أبطئت لمدة أسبوع أو اثنان فهو ليس بنهاية العالم. كُن صادقاً فقط: اشرح لعملائك بأنك تعاني من بعض الآلام المتزايدة، هم قد لا يُثارون ولكنهم سيقدرون الصراحة.

ولكن الحد الأدنى : بان تصنع قراراتك في الوقت المناسب، عندما تحتاج للتعمق في المعلومات الحقيقة .. في هذه الأثناء، ستكون قادر على إغْداق الانتباه على الأشياء التي تتطلب عناية فورية.

استأجر العملاء الملائمين

جِدْ السوق الرئيسية لبرنامجك وركّز عليهم فقط

العميل ليس مصيباً دائماً. والحقيقة أنت الذي يتحتم عليك تصنيف من المحق ومن المخطئ بحق برنامجك. الخبر الجيد هو إن الانترنت جعلت إيجاد الأشخاص المناسبين أسهل من أي وقت مضى .

إن حاولت إرضاء كل شخص، لن ترضي أي شخص

عندما بنينا Basecamp ركزنا سوقنا على تصميم الشركات. بتقليص سوقنا في هذا المجال، جعلناه أكثر مناسبة لجذب العملاء المتحمسين اللذين يقدرون المنتج. اعرف من المقصودين ببرنامجك وركّز على إرضائهم.

أفضل مكالمة صنعناها

قرار حملة مراقبة صارمة في سوق تصميم مواقع الويب كانت أفضل مكالمة صنعناها. لقد سمحت لنا للتمييز بسهولة بأي الميزات المفيدة والمهمة، وأي الميزات التي يجب حذفها. لم نقم بجذب عملاء أكثر باستهداف اصغر مجموعة من الناس فقط، هؤلاء العملاء لديهم حاجات متشابهه تجعل عملنا أكثر سهولة. هناك أطنان من المميزات في حملة المراقبة التي تكون عديمة الفائدة لأي شخص ما عاد مصمم الويب.

أيضاً، فان التركيز على سوق رئيسية يجعل انتشار سمعة برنامجك أكثر سهولة. والآن بعدما عرفنا جمهورنا يمكننا أن نعلن أين يمكنهم أن يترددون مباشرة (online)، ننشر المقالات التي قد تشّد انتباههم، وعموماً نبني جمهور حول منتجنا.

—ديفد قرينر(David Greiner)، مؤسس، ، حملة المراقبة

احسبْ لاحقاً

لا يمكنك قياس المشكلة بعد.

"هل أقيس برنامجي عندما يبدأ الملايين من الناس في استخدامه؟ "

هل تعرف أمراً؟ انتظر حتى يحدث ذلك حقيقة. إذا حصلت على عدد هائل من المستخدمين الذين يزيدون من تحميل نظامك فاهتف ابتهاجاً بذلك! هذه إحدى المشاكل الجيدة التي تتوقف عندها، وفي الحقيقة أن الغالبية الساحقة من برامج الويب لن تصل إلى تلك المرحلة، وحتى إن بدأت في الحصول على تحميل زائد فهي ليست قضية كبيرة، سيكون لديك الوقت لتعديل المشكلة والاستجابة لها. إضافةً، سيكون عندك عالم حقيقي من البيانات و الروابط بعد الإطلاق والتي تساعدك في فهم المناطق التي تستلزم براعة ومهارة.

على سبيل المثال، شغّلنا Basecamp على خادم (server) وحيد في السنة الأولى، ولأننا بدأنا بهذا الإعداد البسيط فإننا أخذنا أسبوع واحد في التنفيذ، ولم نبدأ بمجموعة من 15 خادم أو قضينا شهورا من القلق حول القياسات و التفكير في الحسابات.

هل عانينا من أيه مشاكل؟ قليلاً؛ و أدركنا بان اغلبها قد أخافنا مثل التباطؤ القصير والتثاقل، حقيقة لم يكن ذلك تعامل حسن مع عملائنا. طالما انك تبقى العملاء في الحلقة، وتكون صادقاً حول الموقف فإنهم حتما سيتفهمون . عند التفكير فيما حدث بالسابق، إننا مسرورون جداً بأننا لم نؤخر الانطلاق لأشهر كي نتحقق من " التشغيل المثالي".

في البداية، اصنع منتج ثمين و صلب حسب أولوياتك بدلاً من أن تتوجس حول القياسات والخادمات (servers)، **أنشئ برنامج رائع ثم اقلق حول ماذا ستعمل عندما يصبح ناجحاً جداً.** وإلا فانك قد تفقد طاقة، وقت، مال لتشغيل شئ قد لا يحدث أبداً.

صدق أو لا تصدق، المشكلة الكبيرة ليست في القياس والحساب بل في الوصول للنقطة التي يجب أن تحسبها ، بدون المشكلة الأولى لن يكون عندك المشكلة الثانية.

على أيه حال يجب عليك أن تعاود الزيارة

في الواقع أن كل شخص عنده قضايا لحسابها، ولا يمكن لأي شخص يتعامل مع هذا البرنامج الذي يخدم ما بين صفر إلى بضعه ملايين مستخدم أن لا يعود كل ثانية تقريباً لزيارة التصميم والهيكل .

—دارا اوباسنجو (Dare Obasanjo)، مايكروسوفت (من القياس والبدايات)

اصنع برامج عنيدة

على البرامج أن تتخذ عدة نواحي

يناقش بعض الأشخاص البرامج وهم على غير دراية، فيقولون بان المطورين متغطرسين بسبب تحديد ميزات برنامج ما أو لتجاهلهم طلبات ميزة معينه، ويقولون بان البرامج يجب أن تكون مرنة بقدر الإمكان.

نحن نعتقد بان هذا كلام فارغ، أفضل برنامج هو الذي يمتلك رؤية، وأفضل برنامج هو الذي يتخذ عدة نواحي. عندما يستخدم شخص ما برنامجاً فهو لا يبحث عن الميزات فقط؛ انه يبحث عن الرؤية. قرر ما هي رؤيتك واعمل بها.

وتذكر إذا لم تعجبهم رؤيتك، هناك العديد من الرؤى لأشخاص آخرين ، لا تقصد مطاردة أناس لن تجعلهم أنت سعداء.

كمثال عظيم، تصميم wiki الأصلي .. ألغت إدارة وأصدقاء wiki العديد من الميزات التي أُعتبرت تكامل المستند التعاوني في الماضي، فبدلاً من أن يُنسب كل تغيير في المستند إلى شخص محدد؛ أزالوا معظم صور الكُتاب.، وجعلوا المحتوى بأنا اقل و وقت اقل، وقرروا بأنه ليس مهم من كتب المحتوى أو متى كُتب مما أدى إلى اختلاف كبير .. هذا القرار عزز الشعور المشترك بين المجتمع وكان المفتاح الرئيسي في نجاح Wikipedia .

برامجنا تتبع طريق مشابه؛ لا تحاول جعل كل الأشياء لكل الناس، هي لديها موقف وهي تبحث عن عملاء هم في الحقيقة شركاء وهي تتكلم مع أشخاص يشتركون في رؤيتنا . أنت إما من داخل الحافلة أو من خارجها.

الفصل الخامس: إختيار الخصائص

النصف، ليس النصف الأحمق

ابنْ نصف المنتج، ليس نصف المنتج الأحمق

احذر من "كل شئ ما عدا فكرة غبية" تقترب من تطوير تطبيقك للويب. احذف كل فكرة مقبـولة تأتي بالجوار لكنها سوف تطير للأعلى بنسخة نصف حمقاء لتطبيقك. ماذا تريد أنت حقاً أن تفعله هو أن تبني نصف المنتج الذي يعارض الحمق !

تساءل عن ما يعد أساسياً بحق. الأفكار الجيدة يمكن جدولتها. و **مهما يكن خذ ما تعتقد أن منتجك يجب أن يكون عليه واقسمه إلى النصف.** افرز الخواص بانخفاض حتى يتبقى لك الخواص الأكثر أهمية فقط، وكررها مرة أخرى.

مع Basecamp، نحن بدأنا مع جزئية الرسائل فقط، كنا نعلم أنها كانت قلب التطبيق، لذا تجاهلنا أي معالم أخرى، كـ قوائم العمل وغيرها من

العناصر من أجل الوقت الحالي. لذا اسمح لنا أن نتخذ قاعدة للقرارات المستقبلية في استخدامات العالم الحقيقي بدلا من التخمين أو الاعتماد على الحدس.

ابدأ بالإيجاز، بتطبيق ذكي مُركز الهدف ثم اسمح له بكسب السيطرة. بعد ذلك يمكنك البدء الإضافة إلى أساس صلب كنت قد بنيته.

إنها غير مهمة فقط

الأساسيات فقط

الإجابة المفضلة لدينا على سؤال "لماذا لم تفعل هذا أو لماذا لم تفعل ذاك؟" هو دائماً: " لأنها غير مهمة فقط" وهذا التقرير يجسد مالذي يجعل المنتج عظيم. اكتشف ما هو المهم فقط واترك البقية.

عندما أطلقنا Campfire سمعنا بعض من هذه الأسئلة من أشخاص فحصوا المنتج لأول مرة:

"لماذا الوقت يُكتب كل 5 دقائق فقط؟ لماذا لا يظهر مع كل سطر محادثة؟" الإجابة: لأنه غير مهم فقط. كم مرة تحتاج لمتابعة محادثة ما بالثانية أو حتى بالدقيقة؟ بالتأكيد ليس أكثر من 95% من الوقت. الظهور كل 5 دقائق كافي لان أي شيء أكثر من المحدد ليس مهماً.

"لماذا لم تسمحوا بالتنسيق الغامق أو المائل أو الملون في المحادثات؟" الإجابة: لأنها غير مهمة فقط. إذا كنت بحاجة للتأكيد على شيء ما استخدم الحروف الكبيرة بالنقر على Caps Lock أو علامتي التنصيص حول الكلمة أو العبارة. هذه الحلول لا تتطلب برامج إضافية، دعم فني، معالجة مستمرة، أو امتلاك دورة تعليمية. علاوة على ذلك، التنسيقات الثقيلة في دردشة نصية بسيطة ليست مهمة.

"لماذا لم تظهروا عدد الأشخاص الكلي في الغرفة عند وقت معين؟" الإجابة: لأنها غير مهمة فقط. أسماء الجميع مُدرجة لذا أنت تعلم من الموجود هنا، لكن ما لفرق الذي يصنعه العدد الكلي إذا كان هناك 12 أو 16 شخصاً؟ إذا كان هذا لا يغير تصرفك، فهو غير مهم فقط.

هل حدوث هذه الأمور لطيفاً؟ بكل تأكيد . ولكن هل هي أساسية؟ هل هي حقاً مهمة؟ كلا .. وهذا هو السبب في تركنا لها. أفضل المصممين وأفضل المبرمجين ليسوا الأفضل مع أفضل المهارات، أو الأصابع الرشيقة، أو اؤلئك اللذين يرقصون على أنغام الفوتوشوب أو بيئة اختيارهم، هم اؤلئك اللذين استطاعوا تحديد ما هو غير مهم فقط. هناك، حيث يصنعون المكاسب الحقيقية.

معظم الوقت الذي تقضيه يضيع في أمور غير مهمة. إذا استطعت فصل العمل عن التفكير في غير المهم؛ فسوف تحقق إنتاجية لا يمكنك تصورها أبداً.

ابدأ مع لا

اجعل الخواص تعمل بجد لتكون مُنفذة

السر في بناء نصف المنتج بدلا من النصف الأحمق للمنتج هو قولك لا.

كل مرة تقول نعم للخواص، سوف تتبنى طفلاً، يجب عليك اخذ رضيعك في سلسلة من الأحداث (مثل التصميم، التنفيذ، الاختبار، وما إلى ذلك) وما أن تخرج الفكرة لحيز التطبيق؛ سوف تلتصق معها. فقط حاول إطلاق الخاصية بعيداً عن العملاء وانظر كيف تحصل على القبول عند انسيابها للخارج.

لا تكن رجل الـ نعم

اجعل كل خاصية تعمل بجد لتكون منفذة، اجعل كل خاصية تثبت نفسها و تُظهر أنها الناجية؛ انه مثل "نادي القتال". ينبغي عليك أن تُحدد الخاصية فقط، إذا كان لديها الاستعداد أن تقف على الشرفة لثلاثة أيام تنتظر السماح لها بالدخول.

هذا هو السبب في انك تبدأ بـ لا. كل طلب لخاصية جديدة تأتي إلينا — أو منا — نوقفه عند لا. نحن نسمع لكن لا نعمل. الاستجابة الأولية هي "ليس الآن"، إذا كان الطلب على الخاصية يُحافظ على مسار العودة عندها نعرف أن الوقت قد حان لأخذ نظرة أكثر تعمقاً. عندئذ، وعندئذ فقط، نأخذ بالاعتبار النظر للخاصية على أنها حقيقة.

وماذا تقول للأشخاص الذين يتذمرون عندما لا تعتمد أفكار خواصهم؟ ذكّرهم لماذا فضلوا التطبيق في أول الأمر. "أنت فضلته لأننا قلنا لا. أنت فضلته لأنه لا يعمل 100 أمر آخر. أنت فضلته لأنه لا يحاول إرضاء الجميع كل الوقت."

"نحن لا نريد ألف خاصية"

قدم ستيف جوبز [Steve Jobs] عرض تقديمي صغير و خاص حول مخزن الموسيقى iTunes لبعض أعضاء الإعلام المستقلون. جملتي المفضلة لذلك اليوم كانت إلى متى يستمر الأشخاص برفع أيديهم قائلين: هل "يعمل [س]؟" ، "هل تخطط لإضافة [ص]؟". أخيراً رد جوبز بقوله: "انتظر انتظر – اجعلوا أيديكم منخفضة. اسمعوا: اعرف أن لديك ألف فكرة لخواص مميزة يمكن أن يمتلكها اي تونز، ونحن كذلك. ولكننا لا نريد ألف خاصية وهذا من شانه أن يجعله قبيحاً. الابتكار ليس قول نعم لكل شيء ، انه عن قول لا للجميع باستثناء الخواص المهمة."

—دريك سيفرس (Derek Sivers)، رئيس و مبرمج، CD Baby و HostBaby
(من قل لا بشكل افتراضي)

التكاليف الخفية

كشف سعر الخواص الجديدة

حتى لو توقفت الخاصية في الماضي عند خطوة "لا" ، فأنت لا تزال بحاجة إلى كشف تكاليفها الخفية.

على سبيل المثال، كن على اطلاع بالخواص التي تعمل حلقات تكرار (أي الخواص التي تؤدي إلى المزيد من الخواص). كان لدينا طلبات لإضافة تبويب الاجتماعات إلى منتج Basecamp، يبدو بسيطاً كفاية حتى تبدأ بفحصه عن كثب. فكر في جميع عناصر تبويب الاجتماعات المختلفة التي قد تحتاجها: المكان، الوقت، الغرفة، الأشخاص، ايميل الدعوة، الجداول الزمنية، مستندات الدعم وما إلى ذلك ناهيك عن أنا كنا في تغيير ترويجي لـ لقطات الشاشة، صفحات الجولة، صفحة المساعدة/ أسئلة وأجوبة، شروط الخدمة وأكثر. وقبل أن تعرف كل هذا؛ فإن فكرة بسيطة يمكن أن تنشئ كرة ثلجية في رأسك تؤدي إلى مقدار من الصداع.

لكل خاصية جديدة، أنت بحاجة إلى ...

- 1. قول لا.
- 2. إجبار الميزة لان تُثبت قيمتها.
- 3. إذا كانت النتيجة بـ "لا" مرة أخرى، توقف هنا. إذا "نعم"، استمر ...
- 4. ارسم الشاشة(الشاشات) / واجهة المستخدم (UI).
- 5. صمم الشاشة(الشاشات) / واجهة المستخدم (UI).
- 6. اكتب الكود (برمجه).
- 7-15. اختبر، اقرص، اختبر، اقرص، اختبر، اقرص، اختبر، اقرص ...
- 16. افحص للتأكد من أن نص المساعدة يحتاج إلى تعديل أم لا.

- 17. تحديث جولة حول المنتج (إذا لزم الأمر).
- 18. تحديث نسخة التسويق (إذا لزم الأمر).
- 19. تحديث شروط الخدمة (إذا لزم الأمر).
- 20. افحص لرؤية ما إذا كان أي من الوعود مكسور أو مهزوز.
- 21. افحص لمعرفة ما إذا كان هيكل التسعير هو المتضرر.
- 22. الإطلاق.
- 23. خذ نفس عميق.

هل يمكنك التعامل معها ؟

ابنْ شيئاً يمكنك إدارته

إذا أطلقت برنامج تابع، هل تمتلك نظام للتعامل مع المحاسبة والتعويضات؟ ربما يجب عليك السماح للأشخاص باستخدام بطاقة الائتمان فقط بدلاً من الكتابة، التسجيل، وإرسال شيك في كل شهر.

هل تستطيع أن تتحمل إعطاء 1 غيغابايت من المساحة مجاناً لمجرد أن قووقل (Google) تفعل؟ ربما يجب عليك أن تبدأ صغيراً على 100 ميغابايت، أو تقدم مساحة كبيرة للحسابات المدفوعة فقط.

الخلاصة: ابنْ منتجات وقدم خدمات باستطاعتك إدارتها، من السهل تقديم الوعود ولكن الأصعب بكثير هو الالتزام بها. تأكد من كل ما تفعلونه هو شيء يمكنكم تحمله فعلياً – تنظيمياً، استراتيجياً و مالياً.

الحلول البشرية

ابنْ برمجيات لمفاهيم عامة وشجّع الأشخاص لإيجاد الحلول الخاصة بهم.

لا تفرض اتفاقيات أو معاهدات على الأشخاص وبدلاً من ذلك اجعل برنامجك عاماً حتى يتمكن الجميع من إيجاد حلولهم الخاصة. أعطي الناس ما يكفي فقط لحل مشكلاتهم بطريقتهم الخاصة وبعد ذلك أخرج من الطريق.

عندما بنينا Ta-da List أغفلنا عمداً الكثير من الأشياء، لا توجد طريقة لتعيين مهام أحدهم، لا توجد هناك طريقة لتحديد التاريخ المناسب، و لا توجد هناك طريقة لتصنيف العناصر، الخ.

كنا نحافظ على الأداة نظيفة ومرتبة عن طريق السماح للأشخاص بالحصول على الإبداع. الأشخاص يحددون كيف سيحلون المسائل بطريقتهم. إذا أرادوا إضافة التاريخ إلى قوائم عناصرهم عليهم فقط إضافة (due: April 7, 2006) في مقدمة هذا العنصر، إذا أرادوا إضافة التصنيف عليهم فقط إضافة [Books] في مقدمة العنصر المستهدف. عمل مثالي؟ لا. مرونة مطلقة؟ نعم.

إذا حاولنا بناء برمجية تعامل عدداً محدود من السينيورهات، سنجعلها تكون اقل إفادة لجميع الحالات عندما لا تنطبق عليها بغض الشواغل!

اعمل أفضل ما تستطيعه مع جذور المشكلة ثم تنحى جانبا، سيجد الأشخاص حلولهم واتفاقياتهم الخاصة ضمن إطارك العام.

انسْ طلبات الخواص

اسمح لعملائك أن تذكرك بما هو مهم

العملاء يريدون كل شيء في وضح النهار، سيغرقونك بطلبات الخواص. افحص منتدى منتجاتنا؛ قسم طلبات الخصائص يعلو دائماً على الأقسام الأخرى بزيادة الحجم.

سوف نسمع التالي: "هذه خاصية إضافية صغـيرة" أو "لا يمكن أن يكون هذا صعباً" أو "هل من السهل إضافة هذا" ينبغي أن تضع هذا في ثواني معدودة" أو "إذا أضفتم هذه سأدفع مضاعفةً" وهلم جرا..

بالطبع نحن لا نخطئ الناس بطلبهم الطلبات، نحن نشجع ذلك ونريد أن نسمع ما يقولون، معظم ما أضفناه إلى منتجاتنا بدأ بكونه طلب لعميل. ولكن، كما ذكرنا سابقاً؛ استجابتك الأولية يجب أن تكون لا. ولكن ماذا تفعل مع كل هذه الطلبات التي تنهال؟ أين ستُخزنها؟ **كيف ستقوم بإدارتها؟ اقرأها فقط وبعد ذلك ألقها بعيداً.**

نعم، اقرأها، القها بعيداً، وأنساها. يبدو ذلك تعجرفاً ولكن ذلك من الأهمية التي ستُبقي المسألة تافهة. هذه الوحيدة التي يجب عليك تذكرها، هذه الوحيدة التي حقا مهمة، لا تقلق بشأن تعقب وحفظ كل طلب يأتي. واسمحْ لعملائك أن يكونوا ذاكرتك، و لو كانت ذاكرتك ضعيفةً حقاً؛ سوف يذكرونك باستمرار حتى انك لن تستطيع النسيان.

كيف وصلنا لهذا الاستنتاج؟ عندما أطلقنا منتج Basecamp لأول مرة قمنا بتتبع كل طلب لخاصية أساسية بـ إنشاء قائمة مهام Basecamp. عندما تكرر الطلب من شخصٍ أخر، حدّثنا القائمة بتأشير أكثر (II أو III أو IIII ، و ما إلى ذلك) . اكتشفنا يوماً ما عندما استعرضنا هذه القائمة وبدأنا العمل بأكثر الطلبات لخاصية ما .

ولكن الحقيقة أننا لم نعد ننظر مرة أخرى، ونحن نعرف مسبقاً الخطوة القادمة التي ينبغي القيام بها، لأن عملائنا يذكروننا باستمرار بنفس الطلبات التي تُطلب مراراً وتكراراً. لم تكن هناك حاجة للقائمة أو الكثير من التحليل لأنه كان يحدث في غالبية الوقت الحقيقي. لا يمكنك نسيان المهم عندما تُذكر به كل يوم.

وأمر إضافي أيضاً: طلب عدد س من الأشخاص لأمراً ما لا يستوجب عليك إدراجه. في بعض الأحيان من الأفضل قول لا والاحتفاظ برؤيتك لهذا المنتج.

تمسك بـالمايوه

اسأل الأشخاص عن مالا يريدونه

معظم أسئلة استبيانات البرمجيات و الدراسات البحثية تُركز حول ماذا يريده الناس في المنتج : "ما هي الخاصية التي تعتقد أنها مفقودة؟" "إذا استطعت أن تضيف شيئاً واحداً، ماذا يمكن أن يكون؟" "ما لذي يجعل هذا المنتج أكثر فائدة لك؟".

وماذا عن الوجه الآخر للعملة؟ لما لا نسال الأشخاص عن مالا يريدونه؟ "إذا استطعت أن تُزيل خاصية واحدة، ماذا يمكن أن تكون؟" "ما لا تستخدمه؟" "ماذا يحصل في طريقك غالباً؟"

الأكثر ليس الحل، في بعض الأحيان المعروف الأكبر الذي تعمله لعملائك هو ترك بعض الأشياء خارجاً.

الابتكار يأتي من قول لا

[الابتكار] يأتي من قول لا لـ 1000 أمر لتتأكد انك لم تحصل على مسار خاطئ أو محاولة عمل الكثير. نحن نفكر دائماً بشان أسواق جديد يمكن أن ندخلها، وهذا لا يحصل إلا بقول لا التي تمكنك من التركيز على الأشياء التي حقاً هي مهمة.

—ستيف جوبز، مؤسس، آبل (من بذرة الإبتكار في آبل)

الفصل السادس: بناء التطبيق

سارع بتشغيل البرامج

سارع بتحويل المشاريع إلى واقع ،وأجعلها تعمل في أسرع وقت

تشغيل البرنامج هو أفضل طريقة لخلق الدافع ، و توحيد جهود أعضاء فريقك ،واستبعاد الأفكار غير العملية. تشغيل البرنامج لابد أن يكون الأولوية رقم واحد من أول يوم.

لا بأس في أن تقلل من العمل و تتجاوز التفاصيل و أن تستخدم طرق مختصرة في بناء البرنامج إذا كان ذلك يؤدي إلى تشغيل البرنامج بشكل أسرع. فور ما تقوم بذلك سيكون لديك رؤية واضحة ودقيقة عن كيفية بناء البرنامج. القصص المصورة Stories، مخططات الصفحات wireframes وحتى خليط أكواد الهتمل كلها أمور تقريبية. إطلاق البرنامج هو الشيء الواقعي.

مع الواقعية يتم تشغيل البرنامج ليقترب الجميع من التفاهم والاتفاق. ويبتعد عن النقاشات المنفعلة سواء حول المخططات أو الفقرات التي لا تخدم البرنامج بأي حال من الأحوال. سندرك في الأخير أن ما كنا نظنه تافهاً هو المهم حقاً.

الأشياء الواقعية تقود إلى ردود فعل واقعية وهذا مايجعلك تصل إلى الحقيقة.

الأشياء الواقعية تقود إلى الاتفاق

عندما تبدأ مجموعة متباينة من الأشخاص في محاولة معرفة واستكشاف الأمور التي تؤدي إلى إنسجامهم فإن ... أرائهم حول ذلك ستميل إلى التقارب إذا تم عمل نموذج لأشياء واقعية . وبطبيعة الحال إذا قاموا بعمل مخططات أو طرحوا الكثير من الأفكار لن يتفقوا. ستصل إلى اتفاق فقط إذا قمت بعمل أشياء واقعية.

—كرستوفر ألكسندر، برفسور في هيكلة المعلومات
(من المفاهيم المتناقضة في تجانس هيكلية المعلومات)

دعها تعمل في أسرع وقت ممكن

لا أذكر أنني قد شاركت في مشروع برمجيات – كبير أو صغير- وكان ناجحاً من حيث الجدول الزمني و التكاليف و الوظائف ، إلا وكان هذا المشروع الناجح لا يستغرق فترة طويلة من التخطيط و النقاش و التطوير غير المتفق عليه. من السهل وأحياناً من الممتع تضييع أوقات ثمينة في اختراع مميزات يتضح في النهاية أنها غير ضرورية أو غير قابلة للتنفيذ.

هذا ينطبق على جميع مستويات التطوير ، تحويل المشروع إلى واقع ثم تشغيله هو تعويذة معقدة. لا تقتصر على المشروع بشكل عام، بل أنه لا بد أن تشمل جميع مكونات التطوير متناهية الصغر منذ البدء في بناء التطبيق.

عندما يكون هناك مهمة لتنفيذ أحد المكونات الرئيسية المتاح ، فإن المطورين بحاجة إلى فهم ومعرفة هل هذا الشيء سيعمل أم لا مع ذلك الجزء من تطبيقهم، وبشكل عام سيحاول –المطور-استخدامه متى ما كان ذلك ممكن. حتى ولو كان التنفيذ غير متقن أو كامل في البداية، فإن التعاون المبكر غالباً ما يؤدي إلى واجهات واضحة المعالم و مميزات تقدم بالضبط ما يريدون عمله.

—مات هامر، مطور ومدير منتجات، كنجا

كرر العمليات إلى أن تحقق الهدف

كرر العمل بشكل مستمر

لا تتوقع نجاح تجربتك من المرة الأولى. دع التطبيق ينمو ليرشدك ، دعه يتشكل ويتطور. في تطبيقات الويب ليس ضرورياً أن تصل الى مرحلة الاتقان . صمم الشاشات و استخدمها ومن ثم اختبرها ،وكرر العملية مرات ومرات.

بدلاً من الإصرار المسبق على أن يكون كل شيء سليماً، فإن تكرار عملية بناء التطبيق تجعلك تتخذ قرارات واعية وبشكل مستمر خلال عملية البناء. بالإضافة إلى ماسبق أنك ستحصل على تطبيق عملي و سريع نظراً لأنك لم تنشد الكمال منذ انطلاقتك. النتيجة ستكون تغذية راجعة وتوجيهات حقيقية لأمور تستحق اهتمامك.

التكرار يؤدي إلى التحرر

لست بحاجة هدف تحقيق الكمال من أول محاولة ، خصوصاً إذا كنت تعرف أنه سيحدث لاحقاً. معرفة أنك ستعيد النظر في المسائل هي أكبر دافع لإنتاج وتجربة الأفكار قبل أن تقرر قابليتها للتطبيق.

قد تكون أذكى مني

قد تكون أذكى مني بكثر.

في الحقيقة من المحتمل جداً إذا كنت مثل معظم الناس وبالتالي مثلي فإنك ستواجهة مشكلة في تخيل مالا تستطيع أن تراه أو تشعر به أو تلمسه.

البشر متميزون في الاستجابة لمتغيرات المحيط البيئي. فنحن نعرف كيف نتصرف سريعاً عندما يهاجمنا حيوان مفترس. و نعرف كيف ننظف المنطقة بعد حدوث فيضان مدمر. لسوء الحظ أننا سيئين في التخطيط للمستقبل وفي فهم نتائج أعمالنا و تحدد أولويات الأمور المهمة فعلاً .

ربما كنت من الأشخاص القلائل الذين يستطيعون إنجاز ذلك بأنفسهم، لا يهم هذا الشيء الآن.

الويب 2 الذي قام على افتراض أن كل شخص يستخدم الويب، سمح للمطورين الأذكياء أن يجعلوا القصور البشري يعمل لصالحهم. كيف؟ من خلال السماح للمستخدمين بقول ما يريدون في حين أن هناك متسع من الوقت –أمام المطورين- لإنجاز شيء ما حيال طلباتهم.

والجملة الأخيرة توضح لماذا يجب عليك أن تطور بهذه الطريقة وكيف أنه يجب عليك أن تروج وتطلق هذا التطبيق مباشرة.

أفهم تطبيقك بشكلٍ جيد. تأكد من أن الأجزاء تعمل مع بعضها البعض. بعدها أطلق البرنامج ثم قم بالتعديل. لا يوجد شخص واحد يعادل ذكائنا مجتمعين .

—سيث قودن، كاتب/رجل أعمال

من الفكرة إلى التنفيذ

انتقل من العصف الذهني إلى المخططات إلى الهتمل HTML إلى البرمجة

وها هي مراحل العملية التي نستخدمها للوصول إلى الواقعية:

العصف الذهني

هو توليد الأفكار. ما الذي سيقدمه هذا المنتج؟ على سبيل المثال في تطبيق Basecamp بدأنا في النظر إلى احتياجاتنا الخاصة. كنا نريد أن نقدم مشروعاً حديثاً. وكنا نريد كذلك أن يتشارك فيه العملاء مع بعضهم البعض. كنا نعلم أن أمثال هذه المشاريع مهمة. وكنا نريد أن تكون الأرشفة مركزية ليتمكن الأشخاص من استعراض الأشياء القديمة. أردنا في تلك المرحلة أن يكون لدينا نظرة عامة و واضحة عن كل ما سيحدث في مشاريعنا. كل هذه الأفتراضات و غيرها كانت بمثابة الأساس لنا.

هذه المرحلة لا تهتم بالتفاصيل بقدر ماهي تدور حول الأسئلة الهامة. ما الذي سيقدمه التطبيق؟ كيف لنا أن نعرف إن كان التطبيق مفيد؟ ما الذي سنقوم به بالضبط؟ هذه المرحلة تدور حول الأفكار الهامة لا مناقشة الأمور الفرعية . في هذه المرحلة لا جدوى من مناقشة التفاصيل.

المخططات الورقية

المخططات اليدوية أدوات سريعة و جيدة و رخيصة ، وهذا بالضبط ماتحتاجه عند البدء . أرسم و شخبط ما تريد سواء مربعات ، دوائر، خطوط. دع أفكارك تنساب على الورق. الهدف من هذه المرحلة هو تحويل التصورات إلى واجهات تصميم غير تفصيلية . هذه المرحلة تجريبية ، لايوجد فيها إجابة خاطئة.

أنشئ شاشات الهتمل HTML

قم بكتابة كود الهتمل لتلك الميزة (أو بشكل أنسب لتلك العملية). قدم شيئاً حقيقياً ليرى كل شخص كيف ستبدو الأمور على الشاشة.

على سبيل المثال في تطبيق Basecamp ، بدأنا بشاشة "أرسل رسالة" ثم شاشة "حرر رسالة" واستمرينا على هذا المنوال.

حتى هذه المرحلة لا تقم بالبرمجة. فقط قم بكتابة خليط من أكواد الهتمل HTML و CSS . أما التنفيذ يأتي لاحقاً .

أبدأ بالبرمجة

عندما يبدو لنا الخليط السابق من أكواد HTML و CSS جيداً و يقدم الوظائف الضرورية بشكل كاف ، ننطلق للبدء في البرمجة.

أثناء عملك في المراحل السابقة حاول أن تبقى مرناً و توقع أن تقوم بالإعادة أكثر من مرة. في كثير من الأمور، يجب أن لا تتردد في التخلص من أي مرحلة قمت بإنجازها بشكل سيء ،لتعيد العمل بها من جديد. في هذه المراحل من الطبيعي أن تقوم بالإعادة أكثر من مرة.

تجنب كثرة الخيارات

حدد التفاصيل الدقيقة بدلاً من أن يقوم بها عملائك

أحياناً تواجه قراراً صعباً: كم عدد الرسائل التي يجب إضافتها في الصفحة الواحدة؟ ستكون في البداية ميّال لقول "الندع عددها حسب التفضيل ، حيث يستطيع الأشخاص اختيار 25 ، 50 أو 100" . مجرد اتخاذ قرار سريع، وهذه أسهل طريقة للخروج من الموقف.

التفضيلات هي وسيلة لتجنب اتخاذ القرارات الصعبة

بدلا ً من استخدام خبرتك لتحديد المسار الأنسب لتطبيقك تركت ذلك في أيدي العملاء. قد تعتقد أنك تصنع بهم معروفا ً ،ولكن في الحقيقة أنت تضع أمور تشغلهم (ومن المحتمل أنهم مشغولون بما فيه الكفاية). شاشات التفضيلات والتي تحتوي على عدد لا نهائي من الخيارات هي عبء على العملاء. لا ينبغي للعملاء التفكير في تفاصيل التفاصيل – لا تضع العبء عليهم في أمورٍ هي من مسؤولياتك.

كذلك التفضيلات سيئة لأنها تنتج برمجيات كبيرة الحجم، فيها الكثير من الخيارات التي تتطلب بدورها الكثيرمن الأكواد. وكل هذا يستلزم القيام بالمزيد من الاختبارات والتصاميم. بالإضافة إلى ما سبق سينتهي بك المطاف إلى مجموعة من التفضيلات و شاشات واجهة لم يسبق لك تصورها. هذه الأخطاء تعني أنك لا تعرف عن: تجزئة التصاميم والجداول المقسمة busted tables و قضايا تقسيم الوثيقة strange pagination وغيرها.

اتخذ القرار

اتخذ القرارات البسيطة نيابة عن عملائك. هذا ما فعلناه في تطبيق Basecamp . عدد الرسائل في كل صفحة 25 . وفي الصفحة الرئيسية يُعرض أخر 25 مقال. الرسائل مرتبة حسب الترتيب الزمني العكسي. أحدث خمس مشاريع تظهر في الداش بورد. لا يوجد أي خيارات أخرى. هذه طريقتنا باختصار.

نعم، قد تتخذ قرارا ً خاطئاً ، لكن ما المشكلة في ذلك .حتماً سيشتكي الناس ويخبرونك بذلك لتقوم بالتعديل. الوصول للواقعية هو القدرة على التغير في أسرع وقت.

الخيارات مكلفة

تبين لنا أن الخيارات مكلفة. بطبيعة الحال هناك بعض الخيارات تقدم مزايا هامة ، بل قد تكون مميزات ضرورية في واجهة التطبيق. ولكن كل شيء له ثمن، لذا عليك أن تنظر للجدوى بعناية فائقة. الكثير من المستخدمين والمطورين لا يفقهون ذلك، والنتيجة تكاليف عالية لخيارات عديمة الجدوى. إذا كنت صارما ً في وضع خيارات افتراضية جيدة بدلا ً من إضافة خيارات بشكل عشوائي، حتما ً سيدفع هذا بالواجهة الرسومية إلى الإتجاة الصحيح.

—هافك بننتون، قائد تقني، القبعة الحمراء (المصدر البرمجيات الحرة و واجهات المستخدم الجيدة)

"اتفقنا!"

القرارات أمور مؤقتة لذا أتخذ القرار وانتقل للخطوة التالية

"اتخاذ القرار" فكر بها على أساس أنها المفتاح السحري. اتخاذ القرار يعني أن شيء ما تم إنجازه ،فالقرار تم اتخاذه وتستطيع أن تتقدم. اتخاذ القرار يخلق لديك الدافعية.

لكن مهلا ً ، ماذا لو ألتبست عليك الأمور واتخذت القرار الخاطئ؟ لا عليك ، **أنه مجرد تطبيق ويب وليس عملية جراحية خطيرة**.كررنا مرارا ً، يتوجب عليك مراجعة المميزات و الأفكار عدة مرات خلال عمليات البناء. مهما خططت فأنه من المحتمل أن تقع أخطاء صغيرة. لا تقم بتحليل كل شيء . فهذا يثبط الروح المعنوية و يبطئ من عملية بناء.

بدلا من ذلك ،أعط قيمة أكبر لأهمية التغير والتقدم . كن منسجما ً مع آلية اتخاذ القرارات. اتخذ قراراً سريعا ً و بسيطا ً وإذا أتضح إنه غير عملي قم بالعودة إليه وتغييره.

عليك تقبُل أن القرارات عملية مؤقتة ، وأن الأخطاء ستحدث لامحالة، وحدوثها ليس قضية كبيرة طالما أنك تستطيع تصحيحها وبسرعة. اتخذ القرار لتخلق قوة دافعة ومن ثم انتقل للخطوة التالية.

كن منفذاً

بالنسبة لي من المضحك جداً عندما أسمع أن الناس متحفظين جداً على الأفكار (الأشخاص الذين يريدون مني الموافقة على فكرة عليهم اخباري بالفكرة الأبسط) .

بالنسبة لي ، الأفكار لا قيمة لها مالم تنفذ. التنفيذ هو الأهم. النتيجة هي حاصل ضربهما.

التوضيح:

- فكرة سيئة = -1
- فكرة ضعيفة = 1
- فكرة متوسطة = 5
- فكرة جيدة = 10
- فكرة ممتازة = 15
- فكرة رائعة = 20

- غير قابلة لتنفيذ = $1
- تنفيذ ضعيف = $1000
- تنفيذ متوسط = $10,000
- تنفيذ جيد = $100,000
- تنفيذ ممتاز = $1,000,000
- تنفيذ رائع = $10,000,000

للبدء في عمل تجاري ، تحتاج إلى ضرب الرقمين.

الفكرة الأكثر روعة مع عدم التنفيذ تساوي 20 $. والفكرة الرائعة عندما يتم تنفيذها بمستوى رائع تساوي 20,000,000 $.

لهذا السبب لا أريد سماع أفكار الناس. فأنا لا أهتم حتى أرى التنفيذ.

—ديريك سايفرس، رئيس ومبرمج، CD Baby و HostBaby

اختبر التطبيق في بيئته

اختبر تطبيقك من خلال الاستخدام الواقعي

لا بديل للأشخاص الواقعيين الذين يستخدمون تطبيقك بطرق واقعية. أحصل على بيانات وتغذية مرتجعة واقعية ومن ثم قم بتحسين التطبيق بناء على تلك المعلومات.

الاختبارات الرسمية لقابلية الاستخدام صعبة جداً. كذلك بيئة المعمل لا تعكس الواقع. فإذا قمت بمراقبة شخصاً ما مراقبة لصيقه، ستعرف بعض الأفكار التي تعمل و التي لا تعمل ، ولكن بشكل عام الأشخاص لا يكون أداؤهم جيد أمام الكميرا . عندما تراقب الآخرين، فأنهم يكونون حريصين على عدم إرتكاب الأخطاء — وهذه الأخطاء هي بالضبط ماتبحث عنه.

عوضاً عن ذلك، قم بإطلاق القليل من المميزات التجريبية مع تطبيقك الحقيقي . أجعل الآخرين يستخدمون المميزات التجريبية إلى جانب المميزات الأصلية. هذا من شأنه أن يضع المميزات التجريبية في مسار العمل الواقعي و يعرضها لبيانات أشخاص واقعيين . وعند ذلك ستصل إلى نتيجة واقعية.

علاوة على ماسبق، لا تقم بإطلاق الصورة النهائية و التجريبية منفصلتين عن بعضهما البعض. يجب أن يكونا نفس الشيء. الإصدار التجريبي يُسير الأمور بشكل ظاهري. بينما الإصدار الواقعي مع القليل من المميزات التجريبية، سيحقق المراد.

الكتاب التجريبي

إذا كان المطورون يقلقون بشأن إطلاق الكود ، فإن نشر كتاب يصيب الناشرين والمؤلفين بالرعب. لأنه فور ما تتم طباعة كتاب يصبح من الصعب تغييره. (في الحقيقة الوضع ليس كذلك، لكن تصور أن التقنيات القديمة مازالت باقية في هذه الصناعة) بالتالي سيتحمل الناشرون الكثير من المتاعب (و النفقات) في سبيل أن تكون الكتب "صحيحة" قبل نشرها.

عندما كنت أكتب كتاب التطوير السريع على الويب باستخدام الريلز، كان هناك طلب كامن من قبل المطورين: أعطنا الكتاب الآن ، نريد أن نتعلم الريلز . لكنني كنت ملزم بإتباع سياسة الناشر. كنت أود قول " أنه لم ينته بعد". لكن ضغط مجتمع المطورين والقليل من التشجيع من قبل ديفيد هينمير هانسون جعلني أغير رأيي. لذا قمنا بإطلاق الكتاب بصيغة PDF قبل أن يكتمل بشهرين. والنتائج كانت مذهلة. ليس فقط لأننا قمنا ببيع الكثير من الكتب ، بل حصلنا على تغذية مرتجعة - وصلنا الكثير من التعليقات. قمت بعمل نظام آلي يلتقط تعليقات القراء، وفي النهاية حصلت تقريباً على 850 تقرير أو خطأ مطبعي، أخطاء فنية، واقتراحات لمحتوى جديد. تقريباً كل هذه الأمور وجدت طريقها للتنفيذ في النسخة النهائية من الكتاب.

كان هذا نجاحاً للكل: فأنا تمكنت من تقديم كتاب مطبوع أفضل مما كان سيكون عليه، والمجتمع حصل على مايريد خلال فترة وجيزة. إذا كنت في سباق تنافسي، إطلاق شيء ما مبكرا يجعل الآخرين معك لا مع منافسيك.

—ديف توماس ، المبرمجون الواقعيون

أفعلها بسرعة

- 1. قرر هل هي جديرة بالعمل كي تقوم بعملها، وإذا كانت كذلك:
- 2. نفذها بسرعة – ليس بشكل كامل. فقط نفذها.
- 3. أحفظها. أرفعها على شبكة الويب. ثم أنشرها
- 4. تعرف على انطباع القراء

> بالرغم من أنني أتردد دائماً في إضافة مميزات جديدة للأشياء التي أقوم بعملها، لكن فور ما أقرر في لحظة ما أن هناك شيء جدير بالعمل ،غالباً سيكون على الموقع بعد عدة ساعات من قراري، سيكون هناك أخطاء ولكن تم إطلاقه ، لأدع التغذية المرتجعة توجه تعديلاتي المستقبلية.
>
> —ديرك سايفرس، مبرمج ورئيس موقع، CD Baby و HostBaby

قلص وقتك

جزئه

تقدير الفترة الزمنية في أسابيع أو شهور ضرب من الخيال. في الحقيقة أنت لا تعرف مسبقاً ما الذي سيحدث في المستقبل البعيد.

لذا قلص وقتك. قسم الزمن إلى فترات زمنية صغيرة. فبدلا من أن يكون مشروع يستغرق 12 أسبوعاً، فكر به على أنه عدة مشاريع تستغرق 12 أسبوعاً . بدلاً من افتراض أن المهام تستغرق 30 ساعة أو أكثر ، قسم المهام إلى أجزاء أكثر واقعية يتراوح الجزء مابين 6-10 ساعات . ثم قم بإنجاز مهمة تلو الأخرى.

أيضاً نفس الفكرة تنطبق على المشاكل الأخرى . هل تواجه مشكلة كبيرة جداً لا تستطيع فهمها؟ قم بتقسيمها. قسم المشاكل إلى أجزاء أصغر وأصغر حتى تستطيع استيعابها.

مهام أصغر و جداول زمنية أصغر

مطوري البرمجيات هم أقلية خاصة من المتفائلين: عندما تُعرض عليهم مهمة برمجية، رأيهم "ستكون مهمة سهلة جداً، لن تستغرق الكثير من الوقت على الإطلاق."

كذلك أيضاً، أعط مبرمجة ثلاث أسابيع لإنجاز مهمة كبيرة، ستستغرق أسبوعين ونصف في المماطلة، وأسبوع أخر في البرمجة. وهناك احتمال ألا تقابل النتائج المجدولة مسبقاً الاحتياجات المطلوبة، لأن المهمة حدثت بشكل أكثر تعقيداً مما يجب أن تكون عليه. بالإضافة إلى ماسبق، من يستطيع أن يتذكر ما الذي اتفق عليه الفريق قبل ثلاث أسابيع؟

أمنح مبرمجة فترة قصيرة لكتابة كود صغير لنموذج محدد ،سوف تقوم بعمله وبسرعة، لتكون مستعدة للخطوة التالية.

المهام الصغيرة و الجداول الزمنية الصغيرة أكثر قابلية للإدارة، وهي كذلك تخفي بعض المتطلبات التي تؤدي إلى إرباكك، و تكلف أقل عندما تغير قراراتك أو تتراجع عنها. الجداول الزمنية قصيرة المدى تجعل المطورين مرتبطين بأعمالهم وتمنحهم المزيد من الفرص للاستمتاع بلذة الإنجاز وأسباب أقل لقول "الحمد لله لدي متسع من الوقت لعمل ذلك. و الآن دعني انتهي من ترتيب الملفات الصوتية في مكتبة الـ آي تونز" .

—جينا ترابني، مطورة ويب ومحررة في موقع Lifehacker، دليل الإنتاجية والبرامج

الحقيقية الكامنة

في يوم من الأيام سيحاول شخص ما أن يجبرك على تقديم إجابة دقيقة لسؤال غيبي . سواء كان ذلك بخصوص الموعد النهائي للعمل أو التكلفة النهائية للمشروع أو سعة الحليب اللازمة لردم وادي قراند كانيون – فقط تنفس بعمق: وقل " لا أعرف" .

بعيداً عن الإضرار بمصداقيتك، هذا الشيء يدل على اهتمامك بالقرارات التي تتخذها. فالمسألة ليست أن تقول بضع كلمات لتبدو ذكياً. لابد من إعادة صياغة السؤال كحوار بناء ليتمكن الجميع من استيعابه. من خلال تعلم كيف لتقديراتك أن تكون دقيقة (ولماذا) يمكنكم العمل معاً لبناء فهم مشترك عن الحقائق الكامنة خلف الأمور.

—ميرلن ماين، مؤسس ومحرر في 43folders.com

أبدأ بحل أول مشكلة تواجهك

بالنسبة لي ففي الويب أفضل شيء على الإطلاق هو إطلاق واعتماد خاصية "nofollow" . لم يتحدث عنها أحد مسبقاً . لم يكن هناك مؤتمرات أو لجان يناقش فيها مجموعة من الأشخاص الطبيعة النحوية أو الدلالية لها. لا يوجد RFC يدور حول فكرة بسيطة مخزن في ملف XML يحتوي على 20 سطر، أرغب بقراءة الكثير حوله من أجل كيفية استخدامه وبعد ذلك لا استخدمه لأنني لست متأكد ما إذا كنت قد قمت بعملية التهيئة للإصدار .3 أو 3.3b .

أنها بسيطة وفعالة و توفر خيار للأشخاص الذي يرغبون بخيار – وهذا مهم جداً عند التعامل مع مستخدمي الويب ، والتي لا تراعي المواصفات و لا رغبات الأشخاص.

في بعض الأحيان حل العشرين مشكلة المقبلة ليس بمثل فائدة و حكمة حل أول مشكلة تواجهنا. هذه ليست مجرد انتصارات صغيرة ضد الإزعاجات البسيطة (كل الانتصارات أمام الإزعاجات البسيطة لا شيء)، بقدر ما هو انتصار بالنسبة لنا ، نحن الذين نستمتع بالنتائج البسيطة والسريعة ، والتي أصبحت محل اهتمام مطور الويب.

—اندري توراز، مبرمج ونائب الرئيس في Federated Media Publishing

الفصل السابع: التنظيم

الوحدة

لا تفصل العمليات عن بعضها

كثير من الشركات تقسم عمليات التصميم،التطوير، كتابة نصوص الواجهة الرسومية، الدعم، التسويق إلى عمليات منفصلة. وعلى الرغم من ايجابيات التخصص المهني ألا أنه قد يتسبب في انغلاق العاملين في عالمهم الضيق ، بدلاً عن الرؤية الشمولية لتطبيق الويب.

اجعل فريقك يعمل بطريقة تكاملية ، كي تخلق حواراً تفاعلياً وصحياً خلال بناء التطبيق ، أوجد نظام تصحيح وموازنة ، ولا تدع الأمور تضيع بين النظرية والتطبيق. أجعل كاتب نصوص الواجهة الرسومية يعمل مع المصمم ، وتأكد من أن المطورين مطلعين على استفسارات الدعم الفني.

و من الأفضل تعيين أشخاص متعددي المواهب، لديهم قدرة العمل في عدة مجالات خلال عمليات التطوير. لتكون النتيجة النهائية منتجاً متجانساً بشكل أكبر.

وقت عمل فردي

الأشخاص بحاجة إلى وقت خالي من مقاطعات لإنجاز الأشياء

37signals هي شركة لها فروع في أربع مدن يفصل بينها ثمان ساعات. من بروفو في ولاية يوتا إلى كوبنهاغن في الدنمارك . خمسة منا يفصلهم عنا ثمان ساعات. أحد الآثار الإيجابية لاختلاف الثمان ساعات بيننا هو توفر وقت خاص لكل فرد على حدة.

لا يوجد سوى 4-5 ساعات خلال اليوم نكون فيها جميعاً مستيقظين ونعمل معاً . أما بقية اليوم ففريقنا يكون نائماً بينما ديفيد الموجود في الدنمارك يعمل. بقية الوقت نكون نحن نعمل بينما ديفيد يكون نائم. هذا الشيء جعل نصف اليوم نكون فيه معاً والنصف الآخر يكون كل شخص يعمل على حدة.

هل لك أن تتوقع ما الفترة التي ننجز فيها معظم العمل؟ إنها فترة العمل الفردي (أي عندما يكون كل فرد يعمل على حدة). في الحقيقة هذا ليس غريباً .فالكثير من الأشخاص يفضلون العمل أمافي الصباح الباكر أو في منتصف الليل – أي الأوقات التي لايكون فيها إزعاج .

عندما تكون في فترة زمنية طويلة لا يقاطعك أحد فيه ، ستصل إلى تلك المرحلة التي تكون فيها إنتاجيتك في أعلى معدلاتها والتي نسميها مرحلة الاستغراق في العمل. وهي التي لا تضطر خلالها للتنقل بين مهام مختلفة. ولاتكون فيها مُقاطعاً لكي تجيب على سؤال أو أن تبحث عن شيء ما أو لترسل إيميل أو لتجيب على رسائل فورية. الفترة التي تكون فيها وحيداً هي التي يحدث فيها التقدم الحقيقي.

وصولك لمرحلة الاستغراق في العمل يحتاج وقتاً طويلاً ، وهذا ما يجعل من المقاطعات عدوك اللدود . هذا الشيء يشبه تماماً مايحدث في حالة العينية "حركة العين السريعة" . فمثلاً هذه الحالة لا تحدث لك مباشرة ، بل تذهب للنوم في البداية ومن ثم تحدث لك هذه الحالة . أي مقاطعة تجبرك على أن تخرج من هذه الحالة لتبدأ من جديد. كذلك العينية هي التي يحدث فيها سحر النوم الحقيقي مثلما تكون فترة العمل الفردي هي سر التطوير الحقيقي.

ضع قاعدة في العمل : نصف اليوم هو فترة العمل الفردي. من 10 صباحاً وحتى 2 ظهرا ، لا يستطيع أن يتحدث أي شخص مع الآخر (باستثناء وقت الغداء). أو أجعل أول أو أخر نصف اليوم هي فترة العمل الفردي . وتأكد من أن هذه الفترة مستمرة لتلغي المقاطعات التي تقتل الإنتاجية .

النجاح في فترة العمل الفردي يعني التخلص من إدمان الاتصالات. خلال هذه الفترة نتخلص من المراسلات الفورية، المكالمات الهاتفية ،

والاجتماعات. و نلغي أي رسائل ايميل تتطلب رد سريع. فقط نغلق كل شيء لنستغرق في العمل.

روتين العمل

نعلم أن العاملين في مجال المعرفة يعملون بشكل أفضل عندما يكونون منهمكين في العمل، حيث يكونون مركزين بشكل تام على أعمالهم ومنصرفين تماماً عن البيئة المحيطة. لذا فهم يفقدون أحساسهم بالوقت ، لينتجوا أشياء عظيمة أثناء التركيز المطلق...المشكلة في ذلك أنه يسهل إخراجك من تلك المرحلة. الضجيج، المكالمات الهاتفية، الخروج لتناول الغداء، الذهاب لدقائق معدودة من أجل تناول قهوة ستاربوكس، والمقاطعات خصوصاً مقاطعات الزملاء –جميعها أمور ستقطع عليك استغراقك. إذا توقفت دقيقة واحدة بسبب أن أحد الزملاء يسألك سؤالاً، هذا الشيء سيربك تركيزك بشكل كاف ليجعلك تحتاج إلى نصف ساعة كي تعود لإنتاجيتك مرة أخرى ، لذا فإنتاجيتك الإجمالية في مشكلة جدية.

— جول سبولسكي، مطور برمجيات، Fog Creek Software
(من من أين يحصل هؤلاء الأشخاص على الأفكار "غير الأصلية")

الإجتماعات سموم

لا تعقد إجتماعات

هل أنت فعلاً بحاجة إلى عقد إجتماع؟ تنشأ الإجتماعات عادة عندما يكون مفهوم ما غير واضح بشكل كاف. بدلاً من اللجوء إلى الإجتماع حاول تبسيط المفهوم ليتسنى لك مناقشته وبسرعة من خلال البريد الإلكتروني أو المراسلات الفورية أو تطبيق Campfire . فالهدف هو تجنب عقد الإجتماعات. كل دقيقة تتجنب قضائها في إجتماع ما هي دقيقة يمكنك قضاؤها في إنجاز عملك الفعلي.

لا يوجد شيء أكثر سمية على الإنتاجية من الإجتماعات. و إليك بعض الأسباب :

- الإجتماعات تجزء يوم عملك إلى أجزاء صغيرة ومتنافرة ، مما يؤدي إلى إعاقة مسار العمل.
- عادة ما تدور حول الكلمات و المفاهيم المجردة وليست حول أشياء حقيقية (مثل جزء من كود أو بعض تصاميم الواجهات) .
- غالباً ما تقدم معلومات قليلة وغير مفيدة أثناء اللقاء.
- غالباً ما يكون هناك شخص معتوه على الأقل ، والذي حتماً سيجد طريقه لتضييع وقت كل عضو بكلام فارغ.
- انجرافها بعيداً عن المواضيع الرئيسية أسهل من انجراف السيارات في الثلوج الكثيفة.
- غالباً تكون جداول أعمالها غامضه ، ولا يوجد شخص متأكد تماماً من الشيء الذي سيحدث.
- تتطلب إعداداً شاملاً ، وهذا الشيء نادراً مايفعله الناس.

وعندما تكون مضطراً لعقد اجتماع (وهذا يجب أن يكون حدثاً نادراً) ، نفذ هذه القواعد البسيطة:

- ضع المنبة على 30 دقيقة، و رنين الجرس يكون إعلاناً بانتهاء الاجتماع.
- قم قدر المستطاع بدعوة عدد قليل من الأشخاص.
- لا تبدأ الإجتماع أبداً دون أن يكون لديك جدول أعمال واضح.

لا تكثر من عقد الإجتماعات

هناك الكثير من الإجتماعات، لذا تراجع عن عقد الاجتماعات التي لا تصب في صالح الإنتاجية. أعقد إجتماعاً عندما يكون لديك قضية أعمال هامة تريد مناقشتها وتحتاج مدخلات أو الموافقة أو الاتفاق حولها. ومع ذلك لاتستجب لاقتراح دعوة كل شخص و أتباعه – لا تضيع وقت الأشخاص بأمور غير ضرورية.

— ليزا هانابيرج، كاتبة (من لا تجعل الإجتماعات تتحكم فيك!)

جزئها

في أثناء نمو المشاريع ، إضافة المزيد من الأشخاص يتسبب في تناقص الأرباح . أحد أهم الأسباب هو زيادة عدد قنوات الإتصال. فشخصان يستطيعان الإتصال مع بعضهما من خلال مسار اتصال واحد . و ثلاث أشخاص يحتاجون إلى ثلاث مسارات إتصال و 4 أشخاص يحتاجون 6 . في الحقيقة نمو الروابط بشكل متصاعد هذا من شأنه أن يجعل الإجتماعات و المذكرات تلتهم يوم العمل بأكمله.

الحل واضح : جزء الفرق إلى وحدات صغيرة ومستقلة عن بعضها البعض لتقلل من روابط الاتصالات بينها.

وبالمثل جزء البرامج إلى وحدات صغيرة. وبما أن جزء كبيراً من المشكلة ينشأ من التبعية (متغيرات عامة ، تمرير البيانات بين الدوال، مشاركة العتاد،، وغيرها) حاول أن تجد طريقة تُقسم من خلالها البرنامج لتزيل أو تقلل التبعية بين الوحدات.

—مجموعة جانسل (من أجعلها صغيرة)

ابحث عن الإنتصارات الصغيرة واحتفل بها

أطلق اليوم شيئاً ما

أهم نقطة في تطوير البرمجيات هي الدافعية. الدافعية هي شعور داخلي – إذا لم تكن محفزاً للعمل الذي تقوم به، فإنه من المحتمل أن تكون جودة عملك أقل مما هو مطلوب ، بل الاحتمال الأكبر أن يفشل ذلك العمل.

الدورة الطويلة لإطلاق البرمجيات تقتل الدافعية . فهي تباعد بين الإنتصارات. بمعنى أخرى المكاسب السريعة التي تحتفل بها هي المحفز الكبير. إذا جعلت دورة إطلاق البرنامج طويلة فأنت بذلك تلغي المكاسب السريعة وبالتالي تقتل الدافعية وهذا مايقتل برنامجك.

إذا كنت في منتصف دورة إطلاق برنامج تستغرق عدة أشهر، أجعل كل يوم أو كل أسبوع (أو أسبوعين) فرصة لتحقيق انتصارات صغيرة. أسأل نفسك "ما الذي أستطيع عمله في 4 ساعات؟" ومن ثم قم بعمله. وممكن أن يكون ذلك ...

- ميزة جديدة وبسيطة .
- تحسينات بسيطة لميزة موجودة.
- إعادة كتابة ملفات المساعدة لتقليل العبء على الدعم الفني.
- إزالة بعض الحقول التي لاتحتاجها.

عندما تجد أن الأربع ساعات منحتك مكاسب سريعة ، ستحتفل بذلك. وهذا من شأنه أن يرفع الروح المعنوية ويزيد الدافعية، ويؤكد على أن الفريق يسير في الإتجاة الصحيح.

الفصل الثامن: التوظيف

وظّف بأقل ، وظّف لاحقاًَ

أضف ببطء لتصل بسرعة

ليس هناك حاجة أن يصبح برنامجك كبيراً في وقت مبكر —— أو حتى لاحقاً، و إن كان عندك إمكانية لإدخال 100 من أفضل الأشخاص، فهي لا تزال فكرة سيئة لمحاولة ذلك و توظيفهم جميعاً في وقت واحد. لا يمكن بأي حال من الأحوال أن تجعل هذا العدد من الأشخاص يستوعب ثقافة متماسكة. سيكون عندك تدريب مصدّع، تصادم للشخصيات، حوار سئ و أشخاص يسيرون في اتجاهات متعددة، و أكثر.

لذلك لا تُوظف، حقاً، لا تُوظف أحداً وابحث عن طريق أخر. هل العمل الذي يرهقك ضروري جداً؟ ماذا لو انك لا تفعله؟ هل بإمكانك أن تحل المشكلة بجزء من برنامج أخر أو تغيير الطريقة بدلاً من ذلك؟

جاك ويلتش، مؤسس eg، حينما يُضطر لطرد شخص ما لا يقوم بتوظيف البديل مباشرةً، يريد أن يرى كم يمكن لـ eg أن تستمر بدون ذلك الشخص أو ذلك الموقع. نحن بالتأكيد لا ندْعو إلى طرد الأشخاص لاختبار هذه النظرية، ولكننا نفعل كتفكير جاك: أنت لست بحاجة العديد من الأشخاص كما تعتقد .

إن لم يكن هناك طريق أخر، فكّر ملياً في التوظيف. ويجب عليك أن تعرف تماماً مْن تستأجر، وكيف تقدمهم إلى العمل، و مقدار الجهد المضبوط المتوقع أن يشعروا به.

قانون بروكس (Brooks')

الموظفين الملحقين بمشروع في مراحله الأخيرة يساهمون بشكل كبير في تأخيره

—فريد بروكس

البرمجة و موسيقى موزرات (Mozart's)

أي مبرمج جيد يعمل وحيدا في مهمة ما لا يحتاج للتنسيق أو الاتصال الضروري بأي شخص، ولكن خمسة مبرمجين يعملون على نفس المهمة يجب عليهم التنسيق والاتصال ببعضهم وهذا يستغرق الكثير من الوقت. المشكلة الحقيقة في توظيف الكثير من المبرمجين متوسطي الخبرة بدلاً عن قليل من المبرمجين الجيدين هي : مهما كانت المدة التي يعملون فيها طويلة؛ فهم لا ينتجون شئ جيد كالذي ينتجه المبرمجون العظماء . خمسة من أنطونيو ساليريز (Antonio Salieris) لن يصنعون موسيقى موزرات أبداً إلا أن عملوا في 100 سنة.

—جويل سبولكسي، مطورة برمجيات، Fog Creek Software (من التقط الملاحظات العالية)

تجاهل الإطارات

اعمل مع الموظّفين المتوقعين على قاعدة اختبار في البداية

هو أمر واحد تأخذه بعين الاختبار، السيرة الذاتية أو مثال لكود برمجي أو عمل سابق . في الحقيقة هناك أمر أخر تعمله مع شخص ما، حينما يكون ممكناً، يُعمل " اختبار تجريبي" لأعضاء الفريق الجدد المتوقعين.

قبل أن نستأجر الموظفين نعطيهم مشروع صغير للمضغ أولاً، نرى كيف يعالجون المشروع، كيف يتصلون، وكيف يعملون، وما شابه. العمل مع شخص ما في تصميم أو برمجة بضعة شاشات سيعطيك طنّ من البصيرة حوله . كما انك ستتعلم بسرعة و بلطف هل هو الإحساس الجيد أم لا .

الجدولة قد تكون قاسية لهذا النوع من التدريب، حتى لو كانت عشرين أو أربعين ساعة فهي أفضل من لا شئ. إن كانت النتيجة جيدة أو سيئة فهي ستكون واضحة. وبهذا ينقذ كل من الجانبين أنفسهم من مشاكل عديدة ويخاطر بالاختبار خارج المشروع أولاً.

ابدأْ صغيراً

محاولة اختبار مهمة صغيرة للبدء، لا تقفز في كل أعمالك حالاً . أعطْ [مساعدك الافتراضي [الجديد مشروع اختبار أو اثنان للعمل عليها وانظر كيف يطوره جذرياً. في البداية يكون من السهل جداً أن يبرر المشاكل المحتملة باللون الوردي، لكنك اجعل الأمر واضحاً: هذا تشغيل اختباري.

—Suzanne Falter-Barns، كاتبة/خبيرة الابداع
(من كيف تجد الاتقان وتُحافظ عليه)

الأفعال ، ليست الأقوال

انظر أي التقنيات المحتملة لاستئجارها على البرمجيات المفتوحة المصدر

الطريقة المثالية للتوظيف في المواقع التقنية — تعتمد على الدرجات، السيّر الذاتية وما شابه. — انه لمُضحك في الكثير من الحالات. هل مهم حقاً من أين حصل احدهم على درجته العلمية أو كم معدله التراكمي؟ هل تستطيع حقاً أن تأمن سيرة ذاتية أو مرجع؟

يكون المصدر المفتوح هدية إلى أولئك الذين يحتاجون لتوظيف أشخاص تقنيين. بالمصدر المفتوح؛ تستطيع تعقب عمل شخص ما و برمجياته — جيده و سيئه — خلال فترة زمنية طويلة.

هذا يعني انه بإمكانك الحُكم على الأشخاص بأفعالهم لا بأقوالهم فقط. وبإمكانك اتخاذ قرار يستند على الأشياء المهمة حقاً :

- **نوعية العمل**
  العديد من المبرمجين يستطيعون الكلام والكلام لكن العثرة عندما يأتي وقت العمل والعمل. بالمصدر المفتوح تحصل على تفصيلات صغيرة جداً لمهارات برمجة و خبرة الشخص .
- **النظرة الثقافية**
  البرمجة هي كل شئ عن القرارات، الكثير والكثير منهم. القرارات موجّهه بأولويتك الثقافية، بقيمك، وغاياتك. انظر إلى القرارات المحددة من قبل المختص بالبرمجة، الاختبارات، والمناقشة ثم انظر إلى من تعطي التسابق الثقافي. إذا لم يحصل أي توافق هنا، فكل قرار سيخلق صراع!
- **المستوى العاطفي**
  عند تعريفه؛ تدّخل المصدر المفتوح يَتطلّب بعض الرغبة. وإلا لماذا يقضى هذا الشخص الكثير من الوقت جالساً أمام الشاشة؟ كمية التدّخل المصدري المفتوح تُظهر في اغلب الأحيان كم يهتم هذا الشخص بالبرمجة.
- **اكتمال الربح**
  كل الذكاء، الميول الثقافية اللائقة و الرغبة لا تعني للبرامج الثمينة إذا لم يستطع الشخص عمل المادة المطلوبة. لسوء الحظ؛ الكثير من المبرمجين لا يستطيع ذلك. لذا ابحث عن الحماس لشحنه، وظّف الشخص الذي يستطيع إخراج الباب و يرغب في جعل المهنة العملية أمرٌ ضروري.
- **التنافس الاجتماعي**
  العمل مع شخص فترة زمنية طويلة، وخلال كلٍ من الضغط/الراحة و الارتفاع/الانخفاض، سيظهر لك حقيقة شخصيته، إذا وجدته يفتقر إلى الأساليب والمهارات الاجتماعية؛ رشحه خارجاً.

عندما يتعلق الأمر بالمبرمجين؛ نستأجر فقط الأشخاص اللذين نعرفهم من خلال البرمجية المفتوحة المصدر. ونرى أي تصرف أخر غير مسئول. وظفنا Jamis لأننا تعقبنا إسهاماته ومشاركاته في مجتمع Ruby ، برع في كل النقاط التي ذكرت أعلاه. لم يكن من المهم الاعتماد على العوامل الثانوية منذ أن استطعنا الحكم عليه بالاستناد على الأمور المهمة: نوعية و جودة عمله.

ولا تقلق بشأن النشاطات اللا منهجيه الإضافية فهي ستأخذ مركزاً و رغبة بعيداً عن عمل الموظف اليومي. إنها مثل الكليشة القديمة التي تقول: إذا أردت انجاز أمرٍ ما؛ اسأل الشخص الأكثر انشغالاً لتعرف. جيمس و ديفيد اثنان من المساهمين الأكثر ثقلاً وانشغالاً و ما زالا يستطيعان إدارة 37signals تقنياً . الأشخاص المحبين للبرمجة و التأكد من انجاز العمل هم بالضبط نوع الأشخاص الذين تريدهم في فريقك.

هواية المصدر المفتوح

في الغالب، الذي تريده من توظيف شخص جديد أن يكون هاوي ومحب لما يعمل ، وليس هناك طريقة أفضل لإظهار ذلك من الالتزام بعمل المشاريع المفتوحة المصدر.

—Jarkko Laine، مطور برمجيات
(من للحد من الخطر، وظِف من المصادر المفتوحة)

استفد من الأفراد المحيطين

اختر أشخاص ذوي تعلم سريع ومتعدد الاختصاص أفضل من المختصين .

لن نقوم بتوظيف شخص يكون مصمم للمعلومات فقط، لأنه سيكون مخصص ومحدد أكثر مما ينبغي. مع فريق صغير مثلنا؛ لا يصبح مفهوما ضروريا استجار أشخاص بمثل هذه المهارة المحددة بدقة.

تحتاج الفرق الصغيرة إلى أشخاص لديهم القدرة على ارتداء العديد من القبعات المختلفة، تحتاج إلى مصممين يستطيعون الكتابة، تحتاج إلى مبرمجين يستطيعون فهم التصميم. كل شخص يجب أن تكون عنده فكره حول كيفية تصميم المعلومات (مهما كانت تعني) . يحتاج كل شخص لان يكون عنده تفكير منظم كما أن كل شخص من الضروري أن يكون قادراً على الاتصال مع العملاء.

ومن الضروري لكل شخص أن يكون قادر وراغب على تغيير الطريقة على طول المسار. احتفظ في راسك إن الفرق الصغيرة غالبا ما تحتاج إلى تغيير الاتجاه وتعمل ذلك بسرعة. أنت تريد شخص يستطيع التعلم والتعديل والعطاء أكفأ من شخص يلتصق بالطين ويعمل شيئاً واحداً فقط.

لن تَستطيع تزييف الحماس

اختر شخص سعيد ومتوسط على الشخص المُحبط والعظيم

الحماس هو الخاصية الوحيدة التي لا يمكنك تزييفها. عندما يأتي وقت التوظيف لا تفكر بأنك تحتاج إلى مُعلم أو تقني مشهور. في اغلب الأحيان يكونون كـ الفقاعات الصابونية. أي موظف متوسط وسعيد أفضل من خبير ساخط أو مُحبط .

ابحث عن الشخص المتحمس، شخص تأمنه في انجاز العمل عندما تتركه لوحده. شخص ما عانى في شركة أكبر وأبطئ ويشتاق للعمل في بيئة جديدة، شخص متحمس في بناء ما أنت تبنيه، شخص يكره الأشياء التي أنت تكرهها، شخص يبتهج بتسلق قطارِكَ.

النقاط الإضافية لسؤال الأسئلة

لاحظ أن الموظفين المتوقعين يسالون الكثير من الأسئلة حول مشروعك، المبرمجون المتحمسون يريدون فهم المشكلة بقدر الإمكان ويقترحون بسرعة الحلول الممكنة والتطويرات، مما يقود للعديد من الأسئلة. تفسيرك للأسئلة يكشف أيضا مفهوم أن مشروعك يمكن أن يُطبّقَ بآلاف الطرق المختلفة وهو ضروري للتثبيت الفهم بقدر الإمكان كما تتخيل عمل برنامجك للويب. وكما انك تشرح التفاصيل؛ فانك تطور إحساس الشخص بأنها منافسة ثقافية جيدة.

—إريك ستيفنز، BuildV1.com

مزخرف الكلمات

وظِف كُتاب جيدين

إذا كنت تحاول الانتقاء بين بضعة أشخاص لملئ الوظيفة، دائما اختار الكاتب الأفضل، لا يهم مهمة هذا الشخص سواء مصمم، مبرمج، بائع، مندوب مبيعات أو مهما كان. مهارة الكتابة ستعود بفائدة. يقود الإيجاز في الكتابة والتحرير إلى الفعاليّة، حيث ينشئ كود مُعبر ، تصميم، رسائل الكترونية، رسائل فورية، وأكثر.

هذا لان الكاتب الحسن يكون جيداً في أكثر من استخدام الكلمات. الكُتاب الجيدون يعرفون كيف يتصلون مع الآخرين، يجعلون الأشياء سهلة الفهم والاستيعاب، يستطيعون وضع أنفسهم في ملابس الأخر، يعرفون ما لذي ينبغي حذفه، يفكرون بوضوح . وتلك النوعيات التي تحتاجها أنت.

العقل المنظم

مهارة الكتابة الجيدة هي مؤشر على عقل منظم قادر على ترتيب المعلومات و المتغيرات بطريقة مرتبة ويساعد كذلك (لا يجعل) الأشخاص الآخرين في فهم الأشياء. تنساب الكلمات داخل الكود البرمجي، والاتصالات الشخصية، والرسائل الفورية -الماسنجر- لمن يتعاونون ضمن مسافات بعيدة. إضافة للمزيد من المفاهيم الداخلية مثل الاحترافية والفعّالية.

—دتسين ميتشل (Dustin J. Mitchell)، مطور (منSignal vs. Noise)

تؤدي الكتابة الواضحة إلى التفكير الواضح

تؤدي الكتابة الواضحة إلى التفكير الواضح . أنت لا تعرف ما لذي تعرفه حتى تحاول إظهاره. الكتابة الجيدة مسألة تتعلق بالشخصية، بدلاً من عمل ما هو سهل لك، أعمل ما هو سهل لقارئك.

—مايكل أ. كوفينغتون (Michael A. Covington)، أستاذ علم الكمبيوتر في جامعة جورجيا
(من كيف تكتب بطريقة أكثر وضوحاً، فكراً أكثر وضوحاً، وتعلم المواد المعقدة بطريقة أسهل.)

الفصل التاسع: تصميم البرنامج

الواجهة أولاً

صمم الواجهة قبل أن تبدأ بالبرمجة

الكثير من التطبيقات تتبني مبدأ البرمجة أولاً ، وهذا المبدأ خاطئ. البرمجة هي أثقل مكونات بناء التطبيق، بمعنى أنها الأكثر كلفة والأصعب في التغيير. لذلك أبدأ بالتصميم أولاً .

التصميم بسيط نسبياً . فالمخططات الورقية سهلة التغيير وليست مكلفة. لا تزال تصاميم الـ HTML سهلة التعديل. هذا الشيء لا ينطبق على البرمجة. البدء بالتصميم يمنحك المرونة. أما البدء في البرمجة يقيدك ويلقي على عاتقك بالمزيد من التكاليف.

السبب الأخر للبدء بالتصميم هو أن الواجهة تعتبر منتجك. فما يراه الناس هو ما ستبيعه. إذا قمت بتأجيل عمل الواجهات إلى النهاية فستظهر الثغرات بشكل واضح.

نحن نبدأ بالواجهة لنر كيف يبدو التطبيق من البداية. ونعيد النظر فيها باستمرار أثناء عملية البناء. هل الواجهة مفهومة؟ هل هي سهلة الاستخدام؟ هل تحل المشكلة المطروحة؟ هذه الأسئلة تجيب عليها بشكل صحيح فقط عندما تتعامل مع شاشات حقيقية. البدء في التصميم يبقيك مرناً ويُمكنك من الإجابة على هذه الأسئلة في مراحل متقدمة من عملية البناء.

القلم البرتقالي الذي جعلني أبدأ في موقع Blinksale

فور ما أدركت حجم مأساتي مع برمجيات الفواتير الجاهزة، قررت رسم ما أفضله وما أريد تطبيقه من حلول للفواتير. سحبت القلم البرتقالي لأنه كان الشيء الوحيد القريب مني ذاك المساء، وقمت برسم 75 بالمائة من الواجهات الرسومية خلال ساعات قليلة. عرضتها على زوجتي ريتشل و التي كانت حينها تكوي، سألتها "ما رأيك؟" أجابت بابتسامة "نفذ هذا الشيء. وبجد" .

وعلى مدى الأسبوعين التاليين قمت بمراجعة التصاميم، وبعمل ملفات الـ HTML الساكنة لجميع صفحات الإصدار الأول من موقع Blinksale . لم نقم بعمل مخططات صفحات wireframes بعد المخططات اليدوية للقلم البرتقالي. بفضل البدء مباشرة بعمل تصميم HTML أصبح مشروعنا واقعي وهذا ما جعلنا مستمتعين جداً في المشروع ، بالرغم من أننا في ذاك الوقت لا نعرف ما الذي فعلناه .

بعد اكتمال ملفات الـ HTML ، فاتحنا مطورنا "سكوت" بفكرة موقع Blinksale . ولأننا نملك مسبقاً تصاميم معظم الواجهات استفدنا كثيراً وعلى عدة مستويات. أولاً أنه أعطى سكوت نظرة واقعية عن اتجاهاتنا . لم تكن مجرد أفكار بل كانت شيئاً واقعياً. ثانياً ساعدتنا وبدقة على قياس كم من الجهد والوقت يحتاج سكوت لتحويل هذه التصاميم إلى تطبيق يؤدي وظائفه. عندما يكون لديك مشروع بتمويل شخصي منك، فإنه من الأفضل أن تبادر بالتنبؤ بموازنتك. تصميم الواجهة أصبح مقياسنا لحدود المشروع الأولية. أخيراً تصميم الواجهة كان بمثابة الدليل الذي يذكرنا بتفاصيل التطبيق أثناء تقدمنا في عملية التطوير. عندما كنا نجرب إضافة مميزات جديدة، لم نكن نقول ببساطة "بالتأكيد، دعونا نضيف ذلك!" كان علينا أن نعود إلى التصميم ونسأل أنفسنا أين يمكن لهذه الميزة الجديدة أن يكون مكانها، وإذا لم يكن لها مكان لا تتم إضافتها.

—جوش وليمز ، مؤسس موقع ، Blinksale

التصميم البؤري

أبدأ من قلب الصفحة متجهاً إلى الخارج

التصميم البؤري يركز على الجوهر الحقيقي للصفحة – البؤرة- ومن ثم يتجه إلى الخارج. هذا يعني أنك في البداية تتجاهل الأمور الثانوية: شريط التنقل داخل الموقع/التبويبات، أسفل الصفحة، الألوان، الشريط الجانبي، شعار الموقع وغيرها. أي أن تبدأ من البؤرة وتصمم في البداية الجزء الأكثر أهمية في المحتوى.

أيا كانت الصفحة فإنه من المؤكد لا قيمة لها بدون وجود مركز أو بؤرة لها. على سبيل المثال، إذا قمت بتصميم صفحة تعرض موضوعاً في مدونة، يعتبر الموضوع نفسه هو البؤرة. فلا الفئات الموجودة في الشريط الجانبي، ولا رأس الصفحة الموجود في الأعلى، ولا نموذج التعليق الموجود في الأسفل ، بل جزئية الموضوع نفسه هي البؤرة. بدون هذه الجزئية ، لا تسمى الصفحة "صفحة موضوع" في المدونة .

فقط عندما تنتهي من تلك الجزئية تستطيع أن تبدأ في التفكير بثاني أهم عنصر في الصفحة. ومن بعد ثاني أهم عنصر تستطيع أن تتجه للثالث وهكذا. هذا هو التصميم البؤري.

التصميم البؤري يتحاشى النموذج التقليدي "دعونا نبني الإطار ومن ثم نحشوه بالمحتوى" . ففي تلك العملية، شكل الصفحة يبني ومن ثم يُدرج شريط التنقل داخل الموقع بعدها تضاف "الأمور" التسويقية. و في الأخير يتم البدء بالوظائف الأساسية، فالغرض الأساسي من الصفحة تحشى به المساحة المتبقية و بأي طريقة. العملية التقليدية تتجه بنا إلى الأسوأ فهي تبقي ما ينبغي أن يحتل الأولوية القصوى في النهاية.

التصميم البؤري يقلب تلك العملية التقليدية ليدعك تركز على المهم فعلاً من اليوم الأول. فالأساسيات أولاً و الأمور الثانوية ثانياً. و النتيجة ستكون شاشات أكثر ودية، وتركيز ، وقابلية للاستخدام بالنسبة للعملاء. بالإضافة إلى ذلك، أنها تسمح بالبدء في حوار بين المصمم والمطور مباشرة بدلاً من انتظار اكتمال جميع جوانب الصفحة .

حل الثلاث حالات

صمم للحالات الطبيعية، الفارغة، الأخطاء

في كل شاشة أنت بحاجة إلى التفكير في الحالات الثلاث المحتملة :

- **الطبيعية**
  الشاشة التي يراها الأشخاص عندما يكون كل شيء يعمل بشكل جيد، ويكون تطبيقك ممتلئاً بالبيانات.
- **الفارغة**
  الشاشة التي يراها الأشخاص عندما يستخدمون التطبيق لأول مرة، قبل إدخال البيانات.
- **الخطأ**
  الشاشة التي يراها الأشخاص عندما يكون هناك خطأ.

الحالة الطبيعية سهلة. و هي الشاشة التي ستقضي فيها معظم وقتك. لكن لا تنس أن تستثمر الوقت في الحالات الأخرى أيضاً (أنظر للمقالات التالية لمزيد من المعلومات حول هذه النقطة).

اللوحة الفارغة

حدد التوقعات من خلال التجارب الأولية المدروسة جيداً

تجاهل مرحلة اللوحة الفارغة هو من أكبر الأخطاء التي ترتكبها. اللوحة الفارغة هي أول انطباع يتشكل حول تطبيقك و لا يمكنك إعادة تشكيله ليبدو جيداً ، تأكد من هذا الشيء.

المشكلة لدينا إنه عند تصميم واجهة المستخدم ، في الغالب يتم ملؤها بالبيانات. دائما ما يملئ المصممون القوالب بالبيانات. فكل قائمة وكل موضوع وكل حقل ،كل مكان يكون ممتلئ بالبيانات. وهذا الشيء يعني أن الشاشة تبدو وتعمل بشكل جيد.

من ناحية ثانية، المرحلة الطبيعية في التطبيق هي أن يكون خالياً من البيانات. فعندما يقوم شخص ما بالتسجيل، فإنه حتماً سيبدأ بلوحة فارغة. هذا يشبه المدونة، فالأمر متروك للأشخاص في إنشائها. فالشكل والمظهر العام لا يتحدد حتى يقوم الأشخاص بإدخال بياناتهم: الموضوعات،الروابط، التعليقات، الساعات، معلومات الشريط الجانبي وغيره.

من المؤسف حقاً، أن العميل يقرر ما إذا كان التطبيق جديراً بالاستخدام في مرحلة اللوحة الفارغة (وهي المرحلة التي يكون فيها الحد الأدنى من المعلومات والتصميم و المحتوى من أجل الحكم وبشكل عام على مدى فائدة التطبيق). عندما تفشل في تصميم لوحات فارغة مناسبة ، فإن الأشخاص لن يتعرفوا على تطبيقك لأن كل شيء مفقود.

بالرغم من أن معظم المصممين والمطورين يعتبرون هذه المرحلة أمراً مسلماً به. إلا أنهم لا يقضون الكثير من وقت التصميم على اللوحة الفارغة والسبب في ذلك لأنهم عندما كانوا يطورون ويستخدمون التطبيق كان في حينها ممتلئاً بالبيانات المدخلة لأغراض الاختبارات. لدرجة أنهم لم يواجهوا اللوحة الفارغة. بالتأكيد أنهم قد يكونوا سجلوا الدخول كمستخدم جديد لدقائق معدودة، لكن غالبية وقتهم يصرف للتنقل في تطبيق ممتلئ بالبيانات.

ما الذي يجب أن تضعه في لوحة فارغة مفيدة؟

- استخدمها كفرصة لإدراج دروس سريعة و وصف مختصر ومساعد.
- قدم من خلالها عينات من لقطات الشاشة لتطبيق يكون ممتلئ بالبيانات ليتسنى للأشخاص معرفته (ولماذا يجب أن يستخدموه).
- أشرح كيف لشخص أن يبدأ، وكيف ستبدو الشاشات في نهاية المطاف، وغيرها.
- أجب على الأسئلة الرئيسية التي سيسألها المستخدم في أول مرة: ما هذه الصفحة؟ ما لذي أستطيع فعله الآن؟ كيف ستبدو هذه الشاشة فور امتلائها بالبيانات؟

- حدد التوقعات وساعد في التخفيف من حدة الإحباط و التخوف و الخلط العام.

الانطباع الأولي في غاية الأهمية. إذا فشلت في تصميم لوحة فارغة مدروسة جيداً ، ستنشئ انطباع سلبي (و خاطئ) عن تطبيقك أو خدمتك.

لن تحصل على فرصة ثانية إطلاقاً ...

الجانب الآخر من واجهة المستخدم في نظام تشغيل ماكنتوش التي أعتقد أنها تأثرت بشكل كبير بـ ستيف جوبز هي تنصيب النظام والتشغيل لأول مرة. أعتقد أن جوبز أدرك تماماً أهمية الانطباع الأولي ... وأعتقد أن جوبز نظر إلى التجربة الأولية وفكر، قد تكون مجرد واحدة من ألف تجربة للمستخدم مع الآلة، لكنها الأكثر أهمية، لأنها الخطوة الأولى لألف تجربة وبالتالي فهي التي تحدد توقعاتهم و تشكل تصورهم الأولي.

—جون قروبر، مؤلف و مطور ويب (المصدر مقابلة مع جون قروبر)

التصميم الدفاعي

عند التصميم خذ في اعتبارك أن هناك أخطاء ستقع

دعنا نعترف بذلك: ستظهر أخطاء عند إطلاق التطبيق. مهما بلغت درجة الاهتمام بتصميم التطبيق، ومهما عملت من اختبارات ، سيواجه العملاء مشاكل. لذا كيف ستتعامل مع هذه الأعطال الحتمية؟ من خلال التصميم الدفاعي.

التصميم الدفاعي يشبه القيادة الدفاعية. مثلما يجب على السائقين أن يكونوا محترسين من الشوارع المنزلقة، ومن السائقين الطائشين، إلى أخره من سيناريوهات خطره، فإن بناة الموقع يجب عليهم البحث باستمرار عن أماكن الخلل التي تسبب الارتباك والإحباط للزوار. الموقع الدفاعي الجيد بيده صنع أو تحطيم خبرة العميل.

نستطيع أن نملأ كتاباً متخصصاً بكل الأمور التي نرغب بقولها عن التصميم الدفاعي. في الحقيقة، لدينا ذلك. عنوان الكتاب "التصميم الدفاعي للويب" ، وهو مصدر رائع لأي شخص يرغب بتعلم كيفية تحسين شاشات الخطأ و غيرها من الأمور الهامة.

تذكر: تطبيقك قد يعمل بشكل رائع بنسبة 90% من الوقت. ولكن إذا تخليت عن العملاء في وقت حاجتهم إليك، فمن المرجح أنهم لن يغفروا لك ذلك.

السياق أهم من الثبات

ما يكون منطقياً هنا قد لا يبدو منطقياً هناك

هل تكون الإجراءات على شكل أزارير أم روابط؟ هذا يعتمد على الإجراء نفسه. هل يتم عرض التقويم على شكل قائمة أو جدول؟ هذا يعتمد على مكان عرضه وعلى طول الفترة الزمنية. هل شريط التنقل الرئيسي داخل الموقع بحاجة إلى أن يكون في كل صفحة؟ هل تحتاج شريط البحث العام في كل مكان؟ هل تحتاج نفس الذيل في كل صفحة؟ الإجابة : "الأمر يعتمد على الحالة نفسها" .

هذا ما يجعل السياق أهم من الثبات. لا بأس في أن يكون تصميمك غير ثابت في سبيل أن يصبح أكثر وضوحاً . قدم للناس فقط ما يهمهم. امنحهم ما يحتاجونه عندما يحتاجونه وتخلص مما لا يريدونه. من الأفضل أن تكون على صواب على أن تكون ثابتاً.

التغير الذكي

الثبات ليس ضرورياً. لسنوات كان طلاب واجهة المستخدم ui و خبرة المستخدم ux يُدرسون أن الثبات في الواجهات هو أحد القواعد الجوهرية في تصميم الواجهة. قد ينطبق هذا على البرمجيات ، لكن على الويب هذا الشيء غير صحيح. المهم على الويب أن يستطيع المستخدم في كل صفحة التقدم للخطوة التالية بسرعة وبسهولة .

في كرييتف قوود، ندعو ذلك بـ" التغير الذكي" :وهو التأكد من أن كل صفحة تقدم وبدقة ما يحتاجه المستخدم في تلك المرحلة من العملية. إضافة عناصر تنقل داخل الموقع ، فقط من أجل الثبات و الاتساق مع بقية صفحات الموقع يعتبر أمراً تافهاً.

—مارك هيرست، مؤسس موقع Creative Good ومنشأ Goovite.com
(المصدر نموذج الصفحة)

كتابة نصوص الواجهة هو تصميم الواجهة

كل حرف مهم

كتابة نصوص الواجهة هو تصميم الواجهة. الواجهات الرائعة هي المكتوبة. إذا كنت تؤمن بأن كل بكسل وكل أيقونة وكل نوع خط مهم، عليك أن تؤمن كذلك بان كل حرف مهم. عندما تقوم بكتابة واجهتك، دائماً ضع نفسك محل الشخص الذي يقرأ واجهتك. ما الذي يحتاج لمعرفته؟ كيف يمكنك شرحه بوضوح وإيجاز؟

هل تضع عنواناً للزر بإسم موافق أو حفظ أو تحديث أو جديد أو إنشاء؟ هذه هي كتابة نصوص الواجهة. هل تكتب ثلاث جمل أم خمس؟ هل تشرح بأمثلة عامة أو بالتفصيل؟ هل تضع عنواناً للمحتوى بأنه جديد أو مُحدث أو محدث قريباً أو معدل؟ هل تقول هناك رسائل جديدة: 5 أو هناك 5 رسائل جديدة ؟ هل تقول 5 أو خمس؟ وهل تقول رسائل أو مواضيع؟ كل هذه الأمور هامة.

كذلك لابد أن تتحدث بنفس لغة جمهورك. كونك تكتب تطبيق ويب فهذا لا يعني أن تستخدم المصطلحات التقنية. فكر بعملائك وفكر ما لذي تعنيه هذه الأزارير والكلمات لهم. لا تستخدم الكلمات المركبة أو الكلمات التي لا يفهمها معظم الأشخاص. لا تستخدم اللكنة المهنية، كي لا تبدو محادثة مهندسين مع بعضهما البعض. أجعل الكلمات قصيرة وعذبة. قل ما أنت بحاجة لقوله فقط.

الكتابة الجيدة لكلمات الواجهة تعني التصميم الجيد. من النادر أن يكون هناك تصميم دون كلمات. الأيقونات لها أسماء، نماذج الحقول لها أمثلة، الأزارير لها عناوين، تعليمات خطوة بخطوة أثناء عملية ما، التفسير الواضح لسياستك في الاسترجاع، كل هذه الأمور واجهة تصميم.

واجهة واحدة

أدمج وظائف الإدارة في الواجهة العامة

شاشات التحكم – الشاشات التي تستخدم في إدارة الخيارات و الأشخاص و غيرها- من المحتمل جداً أن تكون سيئة. والسبب يرجع إلى أن غالبية وقت التطوير يصرف على الواجهة العامة .

لإلغاء متلازمة شاشة التحكم السيئة، لا تقم ببناء شاشات مستقلة لتعامل مع الوظائف الإدارية. بدلاً من ذلك، قم ببناء هذه الوظائف الإدارية (مثل التحرير، الحذف، الإضافة) في واجهة التطبيق العامة.

إذا قمت بصيانة واجهتين منفصلتين (مثال: واحدة لعامة المستخدمين وأخرى للمدراء)، فإن كلا الواجهتين ستعاني من مشاكل. في الحقيقة ستدفع ضريبة مضاعفة. لأنك تجبر نفسك على تكرار العمل و هذا يعني أنك تزيد احتمال أن تكون الواجهات سيئة. كلما قل عدد الشاشات التي تهتم بها، تكون النتيجة أفضل.

لا تبني واجهة منفصلة

التطبيق هو كل شيء. أي شيء يمكن تغييره و إضافته و تعديله يمكن إنجازه مباشرة من خلال منطقة الإدارة في التطبيق نفسه.هذا الشيء يمنحنا فرصة رؤية ما يراه عملائنا بالضبط لمساعدتهم أثناء أي مشكلة أو سؤال يواجهونه. عملائنا لا يقلقون بسبب أننا لا نستخدم واجهات متعددة لإنجاز المهام المختلفة. دقيقة واحدة ربما يقضونها في التعامل مع مواعيد العملاء و الدقيقة التي بعدها ربما تكون في إضافة موظف جديد. يجب أن لا يتم إرهاق العملاء بالتنقل بين التطبيقات المختلفة وأن يتم تطوير واجهة موحدة ليكونون قادرين على التكيف مع التطبيق بسرعة.

—أدورد كنيل، مدير التسويق والمبيعات، KennelSource

الفصل العاشر: الأكواد

البرامج الأقل

أبقْ كودك بسيطاً قدر الإمكان

تظن أن مضاعفة الكود ستضاعف تعقيد برنامجك فقط. لكن في الحقيقة، **كل مرة تزيد فيها كمية الكود ينمو برنامجك تصاعدياً ليكون أكثر تعقيداً**. كل إضافة بسيطة، كل تغيير، كل اعتماد على شيء آخر، وكل اختيار له تأثير يعقبه. استمر بإضافة الأكواد بتهور، وقبل أن تعرف، ستنشئ كرة كبيرة من الطين.

الطريق الذي تحارب به هذا التعقيد بـ برامج اقل . البرامج الأقل تعني ميزات اقل، كود اقل، إهدار اقل.

المفتاح هو أن تعيد صياغة أي مشكلة صعبة تتطلب الكثير من البرامج إلى مشكلة أبسط تتطلب برامج اقل بكثير. أنت قد لا تحل المشكلة بالضبط ولكن لا بأس، إن حل 80% من المشكلة الأصلية لأجل 20% من الجهد هو فوز عظيم لك. كما أن المشكلة الأصلية ليست بتلك الدرجة من السوء بحيث تعادل خمس مرات من الجهد لحلها.

البرنامج الأقل يعني بأنك وضعت الكرة الكرستالية جانباً، فـ بدلا من محاولة توقع المشاكل المستقبلية ، تتعامل فقط مع مشاكل اليوم. لماذا ؟ الخوف الذي يمتلكك عن الغد قد لا يتحقق أبداً . لا تُعطل نفسك في محاولة حل هذه القضايا الشبحية.

من البداية ، صممنا منتجاتنا حول مفهوم البرامج الأقل . حينما يكون ممكناً، نُقطّع المشاكل الصعبة إلى أخرى أسهل منها. لقد وجدنا حلول المشاكل السهلة ليست أسهل في التنفيذ أو الدعم فقط ، هي أسهل للفهم وأسهل في الاستعمال. إن كل جزء يساعدنا في كيفية تمييز أنفسنا عن المنافسين ، وعوضاً عن محاولة بناء المنتجات التي تعمل الكثير، نصنع منتجات تعمل أقل وبكفاءة أكثر.

- البرنامج الأبسط أسهل في إدارته.
- البرنامج الأبسط يخفض قاعدة بياناتك مما يعني
- صيانة اقل للعمل (و موظفون اسعد).
- البرنامج الأبسط يقلل تكلفتك من التغيير لذا يمكنك التكيف بسرعة ويمكنك من تغيير رأيك دون الحاجة لتغيير الأساسيات في شفرتك البرمجية.
- البرنامج الأبسط ينتج ثغرات اقل.
- البرنامج الأبسط يعني دعم اقل

الخاصية التي تختارها للإضافة أو الحذف لديها الكثير لتعمله مع البرامج الأقل أيضا . لا تكن خائفاً من قول لا لطلبات الخاصية التي من الصعب عليك عملها، مالم تكن ضرورية جداً . وفّر على نفسك الوقت/ الجهد/ التشويش بتركهم جانباً .

تباطأْ أيضاً، لا تتخذ إجراء على أي فكرة لمدّة أسبوع وانظر هل ما تزال كفكرة عظيمة بعد أن يزول الحماس الأولي. غالباً وقت التنقيع الإضافي يساعد دماغك على الإتيان بحل أسهل .

شجّعْ المبرمجين لعمل عداد للأفكار
تحب أن تسمع: " الطريقة التي أشرت إليها ستستغرق 12 ساعة ولكن هناك طريقة يمكنني عملها ستستغرق ساعة واحدة فقط. لن نعمل س ولكننا سنعمل ص. " دعْ البرنامج يندفع بقوة، اخبر المبرمجين أن يكافحوا من اجل الذي يعتقدون بأنه أفضل أسلوب.

أيضا، ابحث عن أساليب الالتفاف حول كتابة برامج أكثر . هل تُغيّر النسخة على الشاشة لكي تقترح طريق بديل للعملاء بحيث لا يطلبون تغييرات في نموذج البرنامج ؟ مثلاً، هل بالإمكان أن تقترح أن يرفع الأشخاص الصور بحجم معين عوضاً عن تلاعب في الصور على جانب الخادم؟

لكل ميزة تضعها على تطبيقك، اسأل نفسك: هل هناك طريقة تمكننا من إضافتها دون الحاجة إلى هذا البرنامج أو هذا الكود ؟ اكتب الكود الذي تحتاجه فقط بدون زيادة. نتيجة تطبيقك سيكون أكثر طراوة وأقوى .

لا يوجد هناك كود أكثر مرونة من لا كود هناك!

"السرّ" إلى تصميم البرنامج الجيد لا يكون في معرفة ما تضيفه إلى الكود؛ بل يكون في معرفة ما تحذفه خارجاً! انه يكون في تمييز أين تكمن النقاط الصعبة والنقاط اللينة ، ومعرفة أين تترك مساحة/غرفة بدلاً من محاولة الحشر زيادة على التصميم .

—براد ابلتون، مهندس برامج
(مِنْ لا يوجد هناك كود أكثر مرونة من لا كود هناك!)

التعقيد لا يُقاس خطياً بالحجم

إن القاعدة الأكثر أهمية في هندسة البرامج هي على الأقل معرفة : أن التعقيد لا يُقاس خطياً بالحجم ... 2000 برنامج خطي تتطلب أكثر من ضعف وقت التطوير كـ نصف الحجم .

—مجموعة Ganssle (مِنْ احتفظ بها صغيرة)

تفاعل بالسعادة

اختر الأدوات التي تبقيك وفريقك في إثارة وتحفيز .

أي مبرمج سعيد هو مبرمج منتج ، ولهذا نتفاعل بالسعادة و يجب عليك أيضاً. لا تلتقط الأدوات فقط وتمارسها بالاستناد على معايير الصناعة والأداء القياسية، انظر إلى الحاجات المعنوية: هل هناك عاطفة، فخر، علاقة حرفية هنا؟ هل أنت حقاً سعيد بالعمل في بيئة العمل لمدة ثمانية ساعات يومياً ؟

هذا مهم خصوصاً في اختيار لغة البرمجة ، على الرغم من نفاذ البصيرة الشعبية لدى العامة فهم لم ينشئوا المساواة . حول أي لغة يمكن أن تستخدم و حول أي تطبيق ، الاختيار الصحيح يجعل الجهد ليس محتمل أو قابل للاحتمال لكن لطيف و نشيط. يمكن كله بجعل التفاصيل الصغيرة للعمل اليومي أكثر إمتاعاً .

للسعادة تأثير تعاقبي، أسعد المبرمجين يعملون أشياء صحيحة. يكتبون ببساطة، أسطر برمجية مقروءة. يتخذون من النظافة، الوضوح، المقروءة، المُعبر و الأنيق منهجاً لهم. لديهم المُتعة.

وجدنا نعمة برمجية في اللغة Ruby ونقلناه إلى مطورين آخرين في إطار عملنا، تشاركنا كلنا في مهمة تحسين الموظفين وسعادتهم. نُشجّعك في عطاء تلك المجموعة (التي تخصك) دورة.

الخلاصة، فريقك لكي يعمل يحتاج إلى الأدوات التي يحبها ، تكلمنا هنا من ناحية لغات البرمجة ، لكن المغزى ينطبق على التطبيقات، النماذج أو

أي شئ أخر. استعمل الو لاعة التي تجعل الناس متحمسين، أنت ستولد حماس ومنتج أكفأ في النتيجة.

نوع المبرمجين الذين تريد

السبب الأول في رغبتي بناء تطبيقنا بلغة روبي -Ruby on Rails- أنها رائعة جداً، إنتاجية، وصممت بشكل بديع . إنها تميل لجذب نو ع المبرمجين الذي يهتم بترتيب وتصنيف الأشياء. هذا النو ع من المبرمجين هو ما تريده بالضبط لأنهم يصنعون البرامج الجميلة، الأنيقة والمربحة التي تحتاجها لتكسب السوق.

—تشارلز جولي، مدير إداري في برنامج Nisus (مِنْ Signal vs. Noise)

الكود يتحدث

استمع عندما يدفعك كودك البرمجي للأمام

أصغى إلى كودك سيقدم لك الاقتراحات، سيدفعك للأمام ، سيخبرك أين المخاطر تستقر، سيقترح طرق جديدة لعمل الأشياء وهو من يساعدك للتمسك بنموذج البرنامج الأقل .

هل الخاصية الجديدة تتطلب أسابيع من الوقت وآلاف من الأسطر البرمجية ؟ يخبرك كودك من المحتمل أن هناك أسلوب أكفأ . هل هناك طريقة أفضل لبرمجة شيئاً ما في ساعة واحدة بدلا من الطريقة المعتقدة التي تستغرق عشرة ساعات ؟ ثانيةً، كودك سيوجهك. استمع .

كودك يستطيع أن يرشدك إلى المآزق الهينة والخفيفة، انتبه عندما يظهر الطريق السهل. تأكد؛ الخاصية التي تكون سهلة في إنشائها قد لا تكون بالضبط التي الخاصية التي رسمتها في ذهنك ولكن ماذا؟ إذا عملت الخاصية جيداً وأعطتك وقت أكثر لتعمل شيئا آخر ، فاحتفظ بها .

تصنت

لا تقلق حول التصميم، إذا استمعت إلى كودك سيظهر التصميم الجيد. استمع إلى الأشخاص التقنين، إذا هم يعترضون على صعوبة إجراء التغييرات أو ما شابه فخذ هذه الشكاوى بجدية وأعطهم وقت لإصلاح المتغيرات.

—مارتن فاولر، رئيس علماء، ThoughtWorks (مِنْ هل التصميم ميت؟)

إذا أصبح المبرمجين مأجورين لإزالة الكود..

إذا كان المبرمجين مأجورين لإزالة الكود من البرنامج عوضا عن كتابة كود جديد سيكون البرنامج بشكل عام أفضل .

—Nicholas Negroponte، أستاذ التقنيةِ الإعلاميةِ في MIT
(مِنْ And, the rest of the (AIGA Conference) story)

إدارة الديون

ادفع لأجل الكود و صمم "الفواتير"

نحن في العادة نفكر في الديون من ناحية المال لكنه قد يأتي في أشكال أخرى أيضاً . يمكنك بسهولة إنشاء الكود وتخطيط الديون.

الترميم سوية لبعض الأسطر البرمجية التي تكون فعالة أمر جيد ولكنها مقلقة بعض الشئ وأنت تزيد من الرواتب (الديون) ، المغامرة سوية في التصميم جيده بما فيه الكفاية لكنها في الواقع غير جيدة وأنت تعمله مرة أخرى.

من المقبول عمل ذلك من وقت لأخر، في الحقيقة تحتاج اغلب الأحيان إلى تقنية تساعدك في الوصول الحقيقي بأسرع ما يمكن لكامل الأشياء. لكنك ما تزال تحتاج للاعتراف بالدين الذي يجب عليك سداده في وقت ما أو إعادة تصميم الصفحة التي لا بأس بها .

من المقبول عمل ذلك من وقت لأخر، في الحقيقة تحتاج اغلب الأحيان إلى تقنية تساعدك في الوصول الحقيقي بأسرع ما يمكن لكامل الأشياء. لكنك ما تزال تحتاج للاعتراف بالدين الذي يجب عليك سداده في وقت ما أو إعادة تصميم الصفحة التي لا بأس بها .

الأبواب المفتوحة

اخرج بياناتك إلى العالم عن طريق RSS, APIs، ونحوها.

لا تحاول حبس عملاءك ، دعهم يحصلون على معلوماتهم عندما يريدونها و بالطريقة التي يريدونها. لعمل ذلك، يجب أن تتخلى عن فكرة احتكار البيانات و اتركها تعمل عشوائياً . أعطِ الناس دخولاً إلى معلوماتهم من خلال خلاصات RSS . و عرض الـ APIs يجعل المطورين يضيفون على أداتك. عندما تفعل ذلك، أنت تجعل الحياة أكثر سهولة للعملاء و توّسع الإمكانات التي يعملها تطبيقك.

كان الناس يعتقدون أن خلاصات rss مجرد طريق جيد لاستمرار متابعة المدونات أو مواقع الأخبار، الخلاصات لها قوة أكثر من المعتقد ، هي تمنح العملاء أيضا طريق رائع للاطلاع على المحتوى المتغير على التطبيق دون الحاجة للاتصال مراراً و تكراراً. خلال خلاصات Basecamp ، يستطيع العملاء إضافة عناوين url إلى قارئ الأخبار و أعينهم على رسائل المشروع، القوائم التي يجب عملها و المعالم بدون الوصول إلى الموقع بشكل دائم.

APIs تجعل المطورين يزيدون منتجات إضافية على تطبيقك ومن الممكن أن تكون ثمينة. على سبيل المثال، مشروع Backpack يقدم api الذي ينشئ منتجات Chipt تستخدم في بناء قطع لوحة إعدادات نظام التشغيل Mac . القطع تسمح للمستخدمين إضافة وتحرير رسائل التذكير، قوائم الأشياء، وأكثر من ذلك من على سطح المكتب. مدحنا العملاء كثيراً حول هذه القطعة بل أن بعضهم قال أنها السبب الرئيسي في حصولهم على Backpack.

أمثلة أخرى رائعة لشركات تسمح للبيانات أن تظهر بحرية لكي تحصل على التأثير المقابل:

- **خرائط Google المرئية** أحدثت اهتماماً مثير جعل الناس تتجاهل المعلومات من مصادر أخرى (مثل: قوائم الشقة) وتصر على وضع البيان على خريطة Google.
- عرض Linkrolls طريق الناس للحصول على آخر قوائم العناوين **del.icio.us** المعروضة على مواقعهم الخاصة.
- **Flickr** يسمح للأعمال التجارية الأخرى أن تدخل إلى خدماتهم ، فـ يمكن العملاء أن يشتروا كتب فوتوغرافية، ملصقات، نسخ dvd، والطوابع. "الهدف أن يفتحه بالكامل و يقدم لك تشكيلة اكبر من الاختيارات عندما يتعلق بعمل أشياء لصورك " مقولة Stewart Butterfield من Flickr.

أي قطعة تصنع اختلاف

عندما اصدر فريق 37signals مشروع Backpack قبل فترة، انطباعي الأول كان ... آه !

كان حول الوقت الذي أُصدرت فيه منتجات Chipt لـ قطع النمر الخاصة بمشروع Backpack — التي كانت رائعة جداً ليست للتحميل و لا للمحاولة — وأعطيت Backpack نظرة ثانية. النتيجة؟ اختلاف كلي.

حينما تضرب الفكرة، أخرج القطعة، اكتب، و إرسال — انتهينا. يصل البريد الالكتروني بشيء أريد أن القي نظرة عليه؟ ، أخرج القطعة، اكتب،

و إرسال — انتهينا. القطع معلومات فورية زائدة ومتوفرة بسهولة على كل Mac تستعمله. ولان كل شيء يستند على الويب، ليس هناك أي سيطرة أو تزامن — فقط المساهمة المرنة للمحتوى دون الحاجة لتكون قلقاً أين هي تذهب أو كيف سأدخله أنا لاحقاً.

—Todd Dominey، مؤسس ، Dominey Design
(مِنْ المحاولات على Backpack)

الفصل الحادي عشر: اختيار الكلمات

تقارير المواصفات الداخلية غير عملية

لا تكتب تقارير المواصفات الداخلية (1)

هذه المخططات في الغالب ينتهي بها المطاف وهي لم تقدم شيء للمنتج النهائي. والأسباب كالتالي :

تقارير المواصفات الداخلية أوهام

تقارير المواصفات الداخلية لا تعكس الواقع. لأن البرنامج لا يكون واقعياً حتى يقوم المبرمجون ببنائه والمصممين بتصميمه والأشخاص باستخدامه . تقارير المواصفات الداخلية مجرد حبر على ورق.

تقارير المواصفات الداخلية هي محاولة استرضاء الأطراف

فهي تهتم بإسعاد الشخص وإشعاره أنه طرف مهم، وبالرغم من أن هذا الشيء مطمئن ولكنه غير مفيدة . تقارير المواصفات الداخلية لا تهتم في عملية اتخاذ القرارات الصعبة أو عرض التكاليف، وهي الأمور التي يجب الاهتمام بها من أجل تقديم تطبيق رائع.

تقارير المواصفات الداخلية تؤدي إلى اتفاقية مظللة

موافقة مجموعة من الأشخاص على فقرات من نص ما لا يعتبر اتفاقية حقيقية. فكل شخص يمكن أن يقرأ النص نفسه ولكن يفهمه بطريقة مختلفة. حتماً سنسمع في النهاية عبارات مثل: "انتظر ليس هذا ما فهمته" و " هاه ؟ لم نقم بوصفة بهذا الشكل" و "نعم هذا صحيح وقد وافقنا جميعاً عليه حتى أنت قمت بالتوقيع عليه" ، بالطبع تعرف مثل هذه الأمور.

تقارير المواصفات الداخلية تجبرك على اتخاذ أهم القرارات بالرغم من أن لديك القليل من المعلومات

عندما نبدأ في عملية البناء يكون لدينا القليل من المعلومات. وكلما تقدمنا في عملية البناء والاستخدام تعرفنا على التطبيق بشكل أكبر. أن ما يجب أن يحدث هو أن يتم اتخاذ القرارات عندما يكون لديك الكثير لا القليل من المعلومات.

تقارير المواصفات الداخلية تؤدي إلى الإفراط في المميزات

لا يمكننا التراجع خلال مرحلة تقارير المواصفات الداخلية. كذلك لا تكلفة نتحملها عندما نقوم بإضافة نقطة جديدة لأي عمل ننوي القيام به. لذا نحن نحاول إسعاد الآخرين من خلال إضافة مميزات محببة لهم. لينتهي بنا المطاف إلى التصميم لتلك النقاط المكتوبة وليس إلى الأفراد المعنيين. فضلاً عن وصولنا إلى موقع مكتض بالمميزات يحوي 30 تبويب.

تقارير المواصفات الداخلية لا تدعك تتطور و تتغير و تعيد التقييم

أي ميزة في تقارير المواصفات الداخلية تمت الموافقة والتوقيع عليها فأنت ملزم بها حتى ولو أدركت خلال مرحلة التطوير أنها فكرة سيئة. تقارير المواصفات الداخلية لا تتعامل مع الواقعية التي تقول أن كل شيء متغير و لا يظل على حال .

بالتالي ما الذي يجب أن تفعله بدلاً من كتابة تقارير المواصفات الداخلية ؟ اتبع بديلاً أقصر لتنجز شيئاً واقعياً. حاول كتابة صفحة واحدة عن البرنامج الذي تريد عمله. استخدم لغة بسيطة و مختصرة. إذا احتجت أكثر من صفحة لشرحه أعلم أن اللغة معقدة جداً. هذه العملية يجب أن لا تستغرق أكثر من يوم.

بعد ذلك أبدأ في بناء الواجهة – الواجهة ستكون البديل لتقرير المواصفات الداخلية - . ارسم بعض المخططات الورقية السريعة والبسيطة . ثم حولها إلى أكواد باستخدام لغة الهتمل HTML . على خلاف النصوص والتي تكون مفتوحة للتفسيرات المتعددة فإن تصميم الواجهات أرضية مشتركة يتفق عليها الجميع .

عندما يستخدم الجميع نفس الشاشات تختفي الحيرة . ببناء واجهة يستطيع كل شخص النظر إليها واستخدامها والنقر عليها والشعور بها قبل أن تبدأ في القلق بشأن الأكواد. ضع نفسك مكان خبرة العميل قدر المستطاع.

لا تأبه بتقارير المواصفات الداخلية لأنها تجعلك منغلقاً ، وتجبرك على اتخاذ القرارات الرئيسية و الضخمة في وقت مبكر من عملية البناء. إن تجاوز مرحلة تقارير المواصفات الداخلية يبقيك مرناً ويجعل من التغيير عملية غير مكلفة.

(١) تقرير المواصفات الداخلية : تقرير يتضمن وظائف النظام والأهداف والاحتياجات ، ومعايير قياس الأداء وغيرها . ويكون التقرير هو الاتفاقية المبدئية بين فريق تطوير النظم والمستخدمين على مواصفات النظام المطلوبة ، ويستخدم كمرجع لمرحلة التصميم .

تقارير المواصفات الداخلية عديمة الجدوى

تقرير المواصفات الداخلية عديمة الجدوى. لم يسبق لي رؤية تقارير مواصفات داخلية مفيدة ودقيقة معاً .

لقد رأيت الكثير من الأعمال الرديئة جداً وكانت تعتمد على تقارير المواصفات الداخلية. أنها إحدى أسوأ الطرق في كتابة البرمجيات ،و بحكم تعريفها تعني أن البرامج كُتبت لتطابق النظرية وليس الواقع.

—لينوس تورفالدز ، مخترع لينكس (المصدر: لينكس: لينس حول تقارير المواصفات الداخلية)

قاتل المانعين

وجدت أن الأشخاص الذين يصرون على المستندات ذات المتطلبات الشاملة قبل البدء في أي تصميم هم بالفعل "مانعون" يحاولون إبطاء العملية (وهم في الغالب أشخاص لا يقدمون شيئاً لا في التصميم ولا في التفكير الابتكاري) .

أفضل أعمالنا تم إنجازها من خلال مفاهيم كانت في رؤوسنا حول تحسين الموقع .كنا نقوم بعمل نماذج أولية سريعة (ساكنة) ثم نقوم بتغيير التصميم قليلاً ليتم بعدها بناء نموذج أولي حي مع بيانات واقعية. بعد دراسة وتقييم هذا النموذج يصبح لدينا مشروع واقعي و نتيجة جيدة .

—مارك جالبر ، مطور شركة انترانت (المصدر Signal vs. Noise)

تجنب المستندات غير الضرورية

تجنب الأعمال الورقية غير الضرورية

إلغاء تقارير المواصفات الداخلية يعتبر بداية جيدة لكن لا تتوقف عند هذا الحد. لا تكثر من إنتاج الأعمال الورقية في المناسبات المختلفة. لا تقم بعمل أي مستند لن يصبح شيئاً واقعياً .

قم بالبناء لا الكتابة. إن أردت شرح شيءٍ ما حاول أن تعمل له نموذجاً بالحجم الطبيعي و نموذجاً أولياً بدلاً من كتابة مستند مطول. واجهة الاستخدام الحقيقية أو النموذج الأولي سيصبح منتجاً واقعياً. على العكس من ذلك سنجد أن الورق سيؤول في النهاية إلى حاوية النفايات.

إليك هذا المثال : إذا كان مستند مخطط الصفحات wireframe مقدراً له أن يتوقف بحيث لا يصبح أبدا تصميماً فعلياً فلا تتعب نفسك بعمله. وإذا كان يبدأ كمخطط صفحات ثم يتشكل إلى تصميم فعلي قم بعمله.

المستندات التي تعيش منفصلة عن تطبيقك غير مجدية و لن تجعلك تصل إلى ما تريد. كل شيء تعمله يجب أن يتحول تدريجياً إلى شيء واقعي. إذا توقف المستند قبل أن يصبح شيئاً واقعياً ،فأعلم أنه غير ضروري

لا أحد سيقرأه

لا أستطيع عد صفحات مواصفات المنتج أو وثائق متطلبات العمل business requirement والتي كانت ضعيفة وغير مقروءة وغير مستعملة من قبل فريق التطوير سواء أثناء كتابة الأكواد أو مناقشة المشاكل و طرح الأسئلة أو حتى في اختبارات المستخدم. كذلك لقد عملت مع مطورين أمضوا ساعات طويلة في الكتابات المطولة وفي رسائل إلكترونية وصفية ومستندات معايير الترميز والتي هي الأخرى لم تقرأ.

تطوير تطبيقات الويب لا يقوم على الوثائق الهائلة. تطوير البرمجيات هو عملية متكررة ومتغيرة باستمرار يترتب عليها التفاعل و اتخاذ القرارات السريعة واستحالة التنبؤ بالقضايا الفجائية. كل هذه الأمور يستحيل تسجيلها على الورق.

لا تضع وقتك في كتابة وثائق وهمية و مطولة لن يقرأها أحد. في حقيقة إذا أعطيت منتجك مجالاً كافياً لينمو بنفسه ستجد أنه في نهاية المطاف لا يشبه ما كتبته في تلك الوثائق.

—جينا ترابني ، مطورة ويب ومحررة في لايف هاكر ، موقع متخصص في الإنتاجية والبرمجيات

أخبرني بقصة سريعة

اكتب قصصاً لا تفاصيل

إذا وجدت نفسك بحاجة إلى المزيد من الكلمات لشرح ميزة أو مفهوم ، حاول كتابة قصة مختصرة عن ذلك الشيء.لا تدخل في التفاصيل التقنية فقط أذكر قصة مختصرة. اسردها بطريقة إنسانية كما يحدث في المحادثات الشائعة.

لا حاجة لكتابة مقال. فقط وضح مسار ما حدث. سيكون أمراً رائعاً لو يحتوي سيناريو القصة المختصرة على شاشات قمت بتطويرها.

اهتم التجربة بدلاً من التعلق بالتفاصيل . فكر بشكل استراتيجي لا تكتيكي لأن التكتيكات ستحدث فور البدء في بناء جزء من البرنامج. حتى الآن أنت بحاجة إلى قصة تكون كمحادثة استهلالية لتضعك على المسار الصحيح.

استخدم كلمات واقعية

أضف نصوصاً فعلية بدلاً من عينات نص عشوائية

عينات النص العشوائية (1) صديق موثوق به من قبل المصممين. فهي تساعد في معرفة كيف يبدو التصميم فور إطلاقه وهو ممتلئ بالبيانات. ولكن عينات النص العشوائية قد تكون خطيرة.

عينات النص العشوائية تغير طريقة عرض المعلومات. فهي تقلل من أهمية المحتوى النصي لصالح عنصر التصميم المرئي –أي لصالح شكل النص- بدلاً من أن تحتوي على معلومات قيمة لأي شخص قد يدخل و يقرأ الموقع. عينات النص العشوائية تعني أنك لن ترى التغيرات الحقيقية و التي ستظهر فور إضافة معلومات حقيقية. وتعني كذلك أنك لا تعرف كيف ستبدو النماذج في الموقع بعد تعبئتها ببيانات حقيقية. عينات النص العشوائية حجاب بينك وبين الواقعية .

أنت بحاجة لنسخة حقيقية من المعلومات لتعرف كم يجب أن تكون أبعاد الحقول ، و لترى كيف ستتمدد الجداول أو تنكمش ، ولتعرف كيف سيبدو برنامجك.

قدر استطاعتك استخدم كلمات واقعية ذات مدلولات واضحة. فإذا كان موقعك أو برنامجك يتطلب إدخال بيانات أدخل بيانات حقيقية. أكتب النص الحقيقي ولا تقم باللصق من مصادر أخرى. فإذا كان اسماً أكتب الاسم الحقيقي. وإذا كان المدينة أكتب المدينة الحقيقية. وإذا كان كلمة المرور ومكررة مرتين أكتبها مرتين.

صحيح أنه من السهل الإطلاع فقط على قائمة النماذج وملئ الحقول بكلام فارغ كـ) "شسيسشينمتش" "123عسشيبتشسمي" "سشءى2ض9ث7") من أجل قراءتها بسرعة. لكن هذا الشيء ليس واقعياً ، بمعنى أن عملاءك لن يقوموا بعمل ذلك الشيء. هل من المنطقي أن تسلك الطريق المختصر بينما العملاء مجبرون على الطريق الطويل؟ عندما تدخل معلومات غير حقيقية ،لن تعرف الشعور الحقيقي للعملاء عند تعبئتهم لتلك النماذج .

أفعل ما يفعل عملائك وسوف تفهمهم بشكل أفضل. وعندما تفهمهم بشكل أفضل وتشعر بما يشعرون ، ستبني واجهة أفضل.

(1) عينات النص العشوائي هي نصوص عشوائية يستخدمها المصمم أثناء فترة التصميم ليقوم بتعبئة الأماكن بدلاً من أن يستخدم محتوى حقيقي

عينات النص العشوائي نصوص تافهة

إذ لم تكن لديك القدرة لتخيل ما "قد" يكون عليه المحتوى ، فإن اعتبارات التصميم تكون خاسرة. المعنى يصبح مبهماً "لأنه مجرد نصوص" ، قابلية الفهم تنخفض لأنه لا يوجد شخص يعرف أن هذه النصوص معدة للقراءة. تختفي الفرص بسبب أن عينات النص العشوائي التافهة و المستخدمة بدلاً من المحتوى الحقيقي لا تقدم فرصاً. فالنصوص تكتب صغيرة جداً لأنها ليست للاستخدام وكذلك ليتم توفير مساحات بيضاء جميلة.

ـــتوم سميث ، مطور ومصمم (المصدر أكره عينات النص العشوائية ومستخدميها)

امنح منتجك شخصية مستقلة

ما نوع شخصية منتجك ؟

فكر بمنتجك كشخصية اعتبارية. ما نوع الشخص الذي تريده أن يكون عليها ؟ مؤدب ؟ قاسي ؟ متسامح ؟ صارم ؟ مضحك ؟ جامد ؟ جاد ؟ مرن؟ هل تريد أن تظهر كالمشكوك أو الموثوق فيه ؟ كالعارف بكل الأمور؟ أو كالمتواضع والمحبوب ؟

خلال فترة بناء المنتج ضع في اعتبارك ما اخترته من سمات شخصية . استخدم تلك السمات لتوجيه عملية تصميم الواجهة و كتابة نصوصها وفي وضع المميزات. عندما تقوم بتغيير ما أسأل نفسك هل هذا التغيير يناسب شخصية برنامجك.

لمنتجك صوت ، و يتحدث مع عملائك 24 ساعة يومياً .

الفصل الثاني عشر: الأسعار والإشتراكات

إصدارات مجانية

قدم شيئا ً مجانا ً

العالم متضخم بالمنتجات. ولكي تلفت انتباه الآخرين وسط هذه الضجة قدم شيء مجانيا ً .

الشركات الذكية تعرف أن الهدايا الترويجية طريقة رائعة لجذب العملاء. أنظر لشركة أبل ،فهي تقدم برنامج آي تونز مجاناً لخلق طلب على جهاز آي بود و مخزن موسيقى الآي تونز . بعيدا ً عن الإنترنت ،ستجد أن منافذ البيع بالتجزئة تحذو حذوها. وفي ستار بوكس يقولون أن المشتريات الجديدة نشطت لكل خمس عينات مجانية من المشروبات المقدمة للعملاء. شيء جيد.

بالنسبة لنا ، فإن Writeboard و Ta-da list هي تطبيقات مجانية 100% ،نستخدمها لحمل الأشخاص على استخدام منتجاتنا الأخرى. كذلك دائما ً نقدم إصدارات مجانية –بمميزات محدودة- من جميع تطبيقاتنا .

نريد أن يجرب الأشخاص المنتج و الواجهة و فائدة ما قمنا ببنائه. و فور تعلقهم به من المحتمل جدا ً أن يقوموا بالترقية لأحد الخطط المدفوعة (والتي تسمح بمزيد من المشاريع أو الصفحات أو تقدم للأشخاص الوصول لمزيد من المميزات كمثل رفع الملفات و تشفير البيانات SSL).

وحدات صغيرة الحجم

عمل قطع صغيرة الحجم : التقسيمات المتخصص أصغر العروض لجلب الزبائن إليك. التقسيم الفرعي للمنتج أو للخدمة إلى وحدات صغيرة الحجم ، هو قرار اقتصادي و ممتع.

—بن ماكونل و جاكي هيوبا، كاتبان في مدونة كنيسة العميل
(المصدر ما المقصود بالتبشير بالعملاء؟)

قدم أغنيتك الناجحة

مثل ما يحدث في اللقطات الدعائية للأفلام أو الأغاني الناجحة التي ترسل للإذاعة، فكر في تقديم أغنية من كل ألبوم كنوع من الترويج المجاني تهديه للعالم. هذه الأغنية ستجعل الأشخاص يرغبون بشراء الموسيقى الخاصة بك.

لا تقلق بشأن قرصنة هذه الأغنية . دع الأشخاص يستمعون إليها و ينسخونها و يشاركونها مع الآخرين ، تبرع بها لهم. ليكن لديك ثقة أنه إذا سمعها العالم سيدفعون للحصول على المزيد.

—ديرك سايفر ، ، رئيس و مبرمج موقع، CD Baby و HostBaby (المصدر طرق الترويج المجاني)

سهولة الدخول و الخروج

أجعل التسجيل والإلغاء عملية سهلة .

في تطبيقك أجعل عملية الدخول والخروج سهلة قدر الإمكان.

لو كنت عميلاً أرغب باستخدام تطبيقك ، يجب أن تكون العملية سهلة بالنسبة لي ودون جهد يذكر. ضع زر التسجيل كبيراً و واضحاً وظاهراً في كل صفحة من صفحات موقعك. أخبر الأشخاص عن سهولة العملية كأن تقول : "من التسجيل إلى الدخول فقط دقيقة واحدة! ".

ينبغي أن يكون هناك خيار مجاني ليتمكن العملاء من تجربة التطبيق دون إدخال معلومات البطاقة الائتمانية. بعض منافسينا يطلبون مكالمة هاتفية أو موعد أو كلمة سر من أجل تجربة منتجهم. كيف كان تعاملنا مع هذه الحالة؟ جعلنا كل شخص يستطيع تجربة تطبيقاتنا مجاناً وفي أي وقت.

اجعل نموذج التسجيل قصيراً قدر الإمكان. لا تسأل الأشخاص عن أشياء لا تحتاجها ولا تطرح عليهم نماذج طويلة و مفزعة.

المبادئ نفسها تنطبق على عملية إلغاء التسجيل. يجب أن لا يكون همك هو "احتجاز" الأشخاص داخل منتجك. على الرغم من أننا نحزن عندما يقرر الأشخاص إلغاء حسابهم في Basecamp لكننا لم نجعل عملية الإلغاء مفزعة أو مربكة. "ألغي حسابي" رابط واضح وضوح النهار في صفحة حساب الشخص. يجب أن لا يكون هناك أي بريد إلكتروني يرسل أو نموذج خاص يتم تعبئته أو أسئلة للإجابة عليها من قبل العميل.

أيضاً تأكد أنه باستطاعة الأشخاص الحصول على بياناتهم إذا قرروا الرحيل. في تطبيقاتنا نضمن للعملاء قدرتهم على تصدير رسائلهم و تعليقاتهم إلى ملفات XML وفي أي وقت. أنها بياناتهم ويجب أن يكونوا قادرين على التصرف بها كيفما يشاءون.

هذا الشيء على درجة كبيرة من الأهمية، لأن تمكين الأشخاص من التحكم بمعلوماتهم يبني الثقة. فأنت تمد لهم جسر الوصول إلى جزيرة بياناتهم ، وتسمح لهم بالمغادرة دون عقوبات حينما تتوفر لهم عروض أفضل. وهذا ما يبني الشعور بالود ، أنه الشيء الصحيح الذي عليك القيام به.

الخروج بسهولة

لا تحمل المستخدمين في موقعك ضد رغبتهم . فإذا كانوا يريدون المغادرة دعهم يأخذوا المحتوى الذي انشأوه عندما كانوا في موقعك واتركهم يغادرون .. مجاناً .. أجعل البوابة مفتوحة و زود العملاء بما يحتاجونه لتكون لديهم رغبة في العودة ، بدل من أن يعودوا لأنهم مجبرين.

—تشارلي أو دانيل ، محلل، Union Square Ventures
(المصدر : 10 خطوات لنجاح باهر في شركة ويب 2)

الخدع هي حيل الإنسان العاجز

تجنب العقود طويلة الأجل ورسوم التسجيل و.. الخ .

لا أحد يحب العقود طويلة الأجل و ودفع رسوم الإنهاء مقدماً أو الدفع مرة واحدة لرسوم إعداد الخدمة ، لذا تجنبها . فاتورة منتجاتنا شهرية ، لا يوجد عقود لتوقع عليها و يمكنك الإلغاء في أي وقت وبدون أي غرامة. ولا يوجد أي رسوم لإعداد الخدمة.

لا تحاول إيجاد طرق مخادعة للحصول على المزيد من المال ، بل حاول كسب ما تستحقه.

رصاصة الرحمة

خفف من أثر صدمة الأخبار السيئة من خلال التحذيرات المسبقة وتقديم الاستثناءات للعملاء الحاليين .

تريد إيصال أخبار سيئة كمثل زياد الأسعار؟ حاول أن تسلك طريقة غير مؤذية ، مثل أن تحذرهم مسبقاً وبشكل مستمر. بالإضافة إلى ما سبق ، أمنح العملاء الحاليين بعض الاستثناءات لفترة من الوقت. فهؤلاء الناس هم مصدر دخلك ، لابد أن تشعرهم بالتقدير لا أن تبتزهم.

الفصل الثالث عشر: الترويج

الإطلاق الهوليوودي

أبدأ بالدعاية الترويجية ومن ثم العرض الأولي بعدها أطلق التطبيق

إذا تم إطلاق تطبيق على الملأ ولم يقم شخص بتجربته، هل سيعمل التطبيق ضجة؟ النقطة الأساسية هنا أنه ما لم يكن هناك ضجة إعلامية تسبق إطلاق تطبيقك ، فإن الأشخاص لن يتعرفوا عليه.

لخلق حالة من الضجة والترقب أتبع الأسلوب الهوليوودي في الإطلاق : 1) الدعاية الترويجية 2) العرض الأولي 3) الإطلاق.

الدعاية الترويجية

خلال الأشهر القليلة التي تسبق وقت الإطلاق، أبدأ بنشر التلميحات. عرف الناس بالتطبيق الذي تعمل عليه. ضع شعاراً لتطبيقك، وأكتب في مدونتك عن عملية تطويره. ألق بذور الاستفهامات لدى المتلقين مع بقائك غامضاً قليلاً . بالإضافة إلى ذلك، أنشئ موقعاً على الويب حيث يمكنك جمع عناوين البريد الإلكتروني من الأشخاص المهتمين.

في هذه المرحلة، يجب أن تبدأ باجتذاب الخبراء والمطلعين. هؤلاء الأشخاص متطورين جداً ، وهم صناع الذوق العام على الويب. تعامل مع غرورهم باعتبارهم أفضل الأشخاص في هذا المجال. أخبرهم بأنهم سيحصلون على عرض حصري. إذا بدأت المواقع الأخرى أمثال Boing Boing أو Slashdot أو Digg تضع رابط لموقعك سيزداد زوار الموقع. بالإضافة إلى أن ترتيب صفحتك في جوجل (بيج رانك) سيرتفع.

العرض الأولي

قبل الإطلاق بأسابيع قليلة، أطلق المميزات بشكلها الأولي. أسمح بالاستخدام الخفي لبعض الأشخاص. صف لهم شكل المنتج ،على سبيل المثال تطبيقنا Basecamp وضعنا له لقطات للشاشة و رسائل التذكير الملونة و الأحداث milestones و غيرها من مميزات أخرى.

أخبر الأشخاص عن الأفكار والمبادئ التي ينطلق منها التطبيق. مثلاً في تطبيقنا Backpack قمنا بنشر بيان حول التطبيق قبل إطلاقه. هذا الشيء جعل الناس تتحدث و تتصور التطبيق مسبقاً.

تستطيع كذلك توفير بعض "التذاكر الذهبية" الخاصة ليتمكن الحاصلون عليها من استخدام التطبيق قبل غيرهم. ستحصل على فائدة من جراء وجود أشخاص يختبرون تطبيقك في مرحلة البيتا وكذلك هؤلاء الأشخاص سيشعرون بسعادة غامرة بسبب استخدامهم المبكر للتطبيق.

مرة أخرى نقول، شجع الأشخاص على الاشتراك في الموقع ليكون لديك عدد كبير من عناوين البريد الإلكتروني كي تبادر بإبلاغهم فور ما يتم الإطلاق. عندما حان وقت إطلاق برنامجنا، كان لدينا الآلاف من عناوين البريد الإلكتروني لإرسالها وهذا ساعدنا بشكل كبير في جذب الكثير من الأشخاص.

الإطلاق

مرحلة إطلاق التطبيق، يستطيع الأشخاص فيها الذهاب فعلياً إلى "المسرح" لرؤية تطبيقك. بعد أن تم إرسال رسائل إلى عناوين البريد الإلكتروني

للأشخاص الذين قاموا بالتسجيل ، أطلق موقعك التسويقي بشكله النهائي. اخبر الناس عنه قدر استطاعتك. دع المدونات تضع روابط لتطبيقك. أكتب عن مراحل تقدم تطبيقك: كم عدد الأشخاص المسجلين؟ ما التحديثات /التصحيحات التي قمت بها؟ أظهر رغبتك وعزمك و واظب على ذلك.

الطريق إلى يوم الإطلاق

حالما أصبح Blinksale بالشكل الذي أردناه، بدأنا بمراسلة المختبرين. وهم الأشخاص الذين طلبوا منا الحصول على معلومات حول مشروعنا. هؤلاء هم المشجعون، إذا صح التعبير. إذا كان لديك صلاحية الحديث إلى مجموعة من الأشخاص ، يصبح هؤلاء هم أفضل نقطة للبدء.

الشيء الثاني أننا حصلنا على الأذن للحديث إلى المزيد من الأشخاص حول المنتج. تقريباً قبل إطلاق Blinksale بستة أسابيع قمنا بعمل صفحة في موقعنا للمختبرين وصرحنا فيها أن التطبيق هو أسهل طريقة لإرسال الفواتير من على الإنترنت. الصفحة قدمت معلومات مقتضبة لخلق حالة من الترقب والإثارة ، أما التفاصيل الحساسة ظلت طيء الكتمان. عُرض بوضوح على تلك الصفحة نموذج التسجيل في القائمة البريدية، وهو لا يتطلب الكثير مجرد البريد الإلكتروني (جعلنا العملية سهلة) ليتم إعلام المهتمين فور ما يتم إطلاق التطبيق.

أخبرنا الكثير من الأشخاص أو الكثير من الأصدقاء والزملاء والذين كن نعتقد أن التطبيق سيفيدهم. قام هؤلاء بالكتابة عن الصفحة التجريبية في مواقعهم و مدوناتهم. في غضون أيام، تلقينا الآف الرسائل في قائمتنا البريدية. هؤلاء الأشخاص على درجة كبيرة من الأهمية – نقصد بالأشخاص الذين يسألوننا ليتعرفوا بشكل أكبر على منتجنا و متى يتم إطلاقه بشكل فعلي- .

أخيرا ، قبل الإطلاق بأسبوعين تقريباً قمنا بدعوة بعض الأصدقاء والزملاء والخبراء الصناعيين لمساعدتنا في القيام بالاختبار التجريبي لـ Blinksale . هذا الشيء ساعدنا في تقديم المنتج للأشخاص الذين سيستفيدون من استخدامه وكذلك الذين سيساعدوننا في نشر التطبيق بعد إطلاقه. الجدير بالملاحظة أننا لم نجبر أي شخص على استخدام أو الكتابة عن المنتج. أردنا ببساطة أن ينظر إلى التطبيق و أن يتحدث الأشخاص المعنيين عنه عند إطلاقه. في الختام، إذا كنت بصدد بناء ضجة بهذه الطريقة، من الأفضل أن تكون متأكداً أن منتجك يقدم ما هو متوقع منه. وإلا سيكون منتجك مثل الغيوم غير الممطرة.

عندما حان يوم الإطلاق، أرسلنا رسالة إلى كافة العناوين في قائمتنا البريدية، نُعلم بها أصدقاؤنا المدونين، ونشجع مختبري النسخة التجريبية أن ينشروا آراءهم بتجرد تام. ما أسعدنا هو أن الجهود المبذولة منحتنا مكاسب كبيرة. فبعد وقت قصير من إطلاق التطبيق عشرات الآلاف زاروا موقعنا و ألف من هؤلاء قاموا بالتسجيل من أجل استخدام التطبيق.

— جوش وليمز، مؤسس موقع، Blinksale

موقع ترويجي قوي

أبدأ بالدعاية الترويجية ومن ثم العرض الأولي بعدها أطلق التطبيق

المنتج الرائع هو أفضل أداة ترويجية. فإذا وجد الأشخاص أن التطبيق مفيد حقاً سينتشر ذلك بينهم سريعاً.

أيضاً أنت بحاجة إلى موقع ترويجي جيد. ما الذي يجب وضعه في ذلك الموقع؟ هذه بعض الأفكار :

- **نظرة عامة** : أشرح تطبيقك و فوائده.
- **جولة في الموقع**: دليل إرشادي للأشخاص يقدم مميزات مختلفة.
- **لقطات للشاشة و للفيديو**: وضح كيف سيبدو التطبيق وكيف لهم أن يستخدموه.
- **بيان** : وضح الفلسفات والأفكار التي يتبناها تطبيقك.
- **دراسات الحالة** : قدم أمثلة واقعية للأشياء التي يمكن إنجازه في تطبيقك.
- **الإثارة** : استشهادات مقتبسة من العملاء و النقاد و الصحافة و غيرها.
- **المنتدى** : وفر مكان لأعضاء المجتمع ليتمكنوا من خلاله مساعدة بعضهم البعض.
- **التسعير و الاشتراك**: قم بجذب الأشخاص إلى تطبيقك بأقصى سرعة ممكنة.
- **المدونة** : المدونات تُبقي موقعك محدث بأحدث الأخبار والتلميحات و غيره.

أركب موجة التدوين

التدوين قد يكون أكثر فاعلية من الإعلان (و أرخص بكثير)

الإعلان ينطوي على تكاليف باهظة. وتقييم فاعلية الأنواع المختلفة من الإعلان سيكون أكثر تكلفة من الإعلان نفسه. عندما لا تملك الوقت والمال لاستخدام الطرق التقليدية من الإعلان، فكر بطرق الترويج من خلال المدونة.

أنشئ مدونة ليس لعرض منتجك وحسب بل أيضاً لتقدم من خلالها نصائح و تلميحات وحيل و روابط مفيدة. مدونتنا Signal vs. Noise استطاعت أن تجذب آلاف القراء المميزين خلال أسبوع ، وذلك بفضل قصصنا الشخصية والأجزاء المفيدة والغنية بالمعلومات و المثيرة التي نكتبها بشكل يومي.

لذا عندما حان وقت الترويج لأول منتج لنا Basecamp بدأنا من المدونة. بدأنا نكتب في مدونتنا SvN وانتشر بعدها الخبر بين الجميع. الأشخاص أمثال Jason Kottke و BoingBoingers و Jim Coudal وغيرهم أخرين ممن يملكون مدونات شعبية مشهورة ساعدت في إيضاح الأمور ونجاحها.

تطبيقنا الخاص بإنشاء قوائم المهام اليومية هو مثال أخر يوضح مدى قوة التسويق من خلال المدونة. أطلقنا تطبيقنا (قوائم المهام اليومية)Ta-da Lists من خلال موضوع كتبناه في مدونتنا Signal vs. Noise ، وبعد عدة أسابيع كتبت عنه أكثر من 200 مدونة و قام أكثر من 12000 شخص بالتسجيل لإنشاء قوائمهم الخاصة. أما تطبيق Backpack فقد انتشر بشكل أسرع. فخلال 24 ساعة من الإطلاق قام أكثر من 10000 شخص بالتسجيل.

الإغراء المبكر

أعمل ضجة مقدماً وسيبادر الجميع بالتسجيل

تطرقنا له قبل قليل لكن تحمل التكرار: أطلق بعض أجزاء الموقع و أبدأ بجمع عناوين البريد الإلكتروني بأسرع ما يمكن. اختر اسم للموقع وضع شعار له وقم بكتابة جملة أو اثنتين لوصفه، أو على الأقل ضع بعض التلميحات عن ما الذي سيقدمه تطبيقك. ثم دع الآخرين يعطونك عناوينهم البريدية. وبذلك يكون لديك مجموعة من الأشخاص على أتم الاستعداد وينتظرون إعلامهم عن موعد الإطلاق.

الترويج من خلال التعليم

شارك الآخرين بمعلوماتك

عندما يظهر معلم كمتسابق في البرنامج التلفزيوني الشهير جابردي، يعلق المقدم ألكس ترابك هذه "مهنة شريفة" وهو محق فيما قال. أنه لشيء رائع أن تشرك الآخرين في معلوماتك. **وعندما يكون الأمر متعلقاً بتطبيقك ، ستحقق غرضين: تقوم برد الجميل للمجتمع الذي ساندك و تستطيع كذلك أن تروج لتطبيقك في نفس الوقت.**

التعليم كتقنية ترويجية يعتبر وسيلة ميسرة لنشر أسمك واسم منتجك لعدد كبير من الناس. بدلاً من البيع بطريقة جافة "اشتر هذا المنتج" حاول أن تقدم خدمة قيمة لتحصل على اهتمام الآخرين. فهذا الشيء من شأنه أن يخلق ضجة إيجابية تعجز عنها تكتيكات التسويق التقليدي. الأشخاص الذي تعلمهم سيصبحون ناشرين لمنتجك.

للتعليم أكثر من شكل. أكتب في موقعك عن الحيل و التلميحات التي يرغب بها الأشخاص ليشاركوها مع غيرهم. تحدث في المؤتمرات وأبق حتى بعد انتهائها لتقابل وترحب بالحضور. نظم حلقات عمل ليستطيع الهواة الفضوليين تعلم المزيد و التحدث معك مباشرة. قم بعمل مقابلات مع وسائل النشر. أكتب مقالات تحتوي على معلومات مفيدة تشاركها مع الآخرين. و أكتب كتب ؛)

مثال من تاريخنا هي تقنية التلاشي الصفراء، وهي طريقة ابتكرناها يتم من خلالها تسليط ضوء أصفر دقيق على المنطقة التي تم تغيير محتواها في الصفحة. سبق أن كتبنا عنها مقال في مدونتنا Signal vs. Noise . المقال انتشر بين الناس وعدد من زار الصفحة آلاف و الآلاف (حتى هذه اللحظة يحقق المقال حركة مرور عالية).

المقال عمل على كلا المستويين التعليمي و الترويجي. لقد تعلم الأشخاص درس كما أن الكثير من الأشخاص الذين لم يسبق لهم معرفة منتجاتنا عرفوا عنها. مثال أخر: خلال عملية تطويرنا لإطار الروبي أون ريلز، قررنا أن تكون البنية التحتية مفتوحة المصدر. تبين لاحقاً أنها خطوة موفقة. رددنا الجميل إلى المجتمع، بنينا شعوراً ودياً، نلنا الاعتراف بفريق عملنا، حصلنا على تغذية مرتجعة مفيدة، وبدأنا بتلقي التراقيع و المساهمات من كل المبرمجين في أنحاء العالم.

التعليم له عواقب جيدة، وتستطيع أن تحقق من خلاله منافع متبادلة. فأنت تساعد الآخرين و تحصل على ترقية وظيفية محترمة في آن واحد. بالتالي ما الذي تعرفه والذي يرغب العالم بسماعه؟

تحقيق منافع متبادلة

قسم المقالات والتلميحات في مدونتنا هو من أكثر الأقسام شعبية . مشاركة معرفتنا بالاعتماد على التسويق بالبريد الإلكتروني يضمن أن عملائنا يحصلون على أحدث البرمجيات. إذا تمكن عملائنا من تقديم خدمة أفضل إلى عملائهم، يصبح من المحتمل حصولهم –أي عملائنا- على مزيد من الأعمال، وهذا بدوره يؤدي إلى المزيد من العمل بالنسبة لنا. وبالتالي الكل رابح.

مشاركة معرفتنا وبحرية تامة مع الآخرين ساعدنا لنظهر كخبراء في صناعة البرمجيات وعزز من علاقتنا مع عملائنا الحاليين. يعرف العملاء أننا نهتم بجودة عملهم. و أخيراً حصلنا على حركة مرور عالية موجهة إلى موقعنا من قبل محركات البحث و أصحاب المدونات الذين قاموا بمشاركة مقالتنا مع قرائهم الذين لم يسبق لهم السماع ببرمجيتنا.

—ديفيد قراينر، مؤسس موقع، Campaign Monitor

وجبة من المميزات

أنهم متعطشون لها .... لذا قدمها لهم

المميزات الجديدة أو الممتعة هي أروع طريقة لخلق الضجة حول تطبيقك. مجموعات الاهتمامات الخاصة تعشق الحصول على "وجبة المميزات" قبل غيرهم كي يجربوها ويعرضوها على أفراد المجتمع بطريقتهم الخاصة.

على سبيل المثال، باستخدام الروبي أون ريلز- منصة تطوير جديدة- خلقنا ضجة حول تطبيقنا Basecamp داخل مجتمع المطورين.

عناصر الأجاكس التي استخدمناها في تطبيقاتنا أثارت ضجة كبيرة لدرجة أنها قادت مجلة الأعمال التجارية 2.0 إلى تسمية 37signals "كلاعب رئيسي في الأجاكس" جنباً إلى جنب مع الأسماء الكبيرة مثل جوجل و ياهو و مايكروسوفت و أمازون.

مثال آخر: المدونون أخذوا يدعمون خلاصات تطبيق Basecamp لأنه أول تطبيق أعمال يستخدم الخلاصات RSS.

دمج iCal في تطبيقنا قد يبدو ميزة غير مهمة. ولكن على العكس كتبت عنا عشرات المواقع المتعلق بالماكنتوش والتي من المحتمل إنه لم يسبق لها أن أشارت للتطبيق من قبل.

الفرق الصغيرة تستطيع تطبيق الأفكار الجديدة في البرمجيات. أثناء ما تكون الشركات الكبيرة تتعامل مع العوائق البرواقراطية، تستطيع أنت تطبيق الأفكار الجديدة وبسرعة و أن تلفت الانتباه لاستخدامها.

مسايرة الموضة التقنية طريقة فعالة و رخيصة لخلق الضجة. ومع ذلك لا تقم بإضافة أحدث التقنيات الغامضة من أجل لفت الانتباه. لكن إذا كنت تستخدم شيئاً جديداً وجديراً بالاستخدام، أمض قدما ونفذه في البداية لمجموعات الاهتمامات الخاصة.

تتبع سجلات موقعك

أدرس سجلات موقعك لتتبع مصدر الضجة

أنت بحاجة لمعرفة من يتحدث عن موقعك. أفحص سجلات موقعك لتكتشف مصدر هذه الضجة. من قام بوضع روابط لموقعك؟ من ينتقدك؟ أي المدونات الموجودة في Technorati, Blogdex, Feedster, Del.icio.us, Daypop متابعة لموقعك بحماس؟

تعرف على تلك الأمور ومن ثم أجعل الآخرين يشعرون بذلك. علق في تلك المدونات. أشكر الأشخاص على وضعهم لروابط موقعك. أسألهم إذا ما كانوا يرغبون في أن تدرجهم ضمن القائمة الخاصة ليكونوا من بين أوائل الأشخاص معرفة بالإصدارات المستقبلية، التحديثات، وغيرها. أجمع عبارات الثناء ومن ثم قم بإنشاء صفحة في موقعك باسم "الضجة" . شهادات الآخرين أداة رائعة للترويج لتطبيقك خصوصاً أن ثناء الأطراف الخارجية أكثر موثوقية لمعظم الأشخاص.

حتى لو كانت التعليقات سلبية استمر بالاهتمام بها. أظهر أنك تستمع لهم. أجب على الانتقادات بعناية. مثل أن تقول "نحن نقدر لك ملاحظتك لكن قمنا بعملها بهذه الطريقة بسبب ... " أو "أثرت نقطة جيدة ونحن نعمل عليها" . بذلك ستضعف من تلك الانتقادات وتوجه الاهتمام نحو منتجك. من المدهش كيف للردود المدروسة جيداً في مدونة ما أن تبعثر الأشخاص الرافضين بل أنها قد تحولهم إلى أشخاص داعمين لمنتجك.

الإقناع الداخلي للعملاء

روج لفرص الترقية من داخل التطبيق نفسه

كل شخص يعرف كيف يسعر في الموقع التسويقي. ولكن البيع لا يقف عند هذا الحد. إذا كانت لديك خطة تسعير متدرجة (أو نسخة مجانية من تطبيقك)، لا تنسى أن تخبر الآخرين بفرص الترقية من داخل المنتج نفسه.

أخبر الأشخاص بأن العقبات ستزول من أمامهم إذا ما قاموا بالترقية. على سبيل المثال ، في تطبيق Basecamp لا يمكنك رفع الملفات إذا كان حسابك مجاني. عندما يحاول شخص رفع ملف، لا نكتفي بمنعه، بل نشرح له لماذا خاصية رفع الملفات غير متوفرة ونشجعه على الترقية إلى الإصدار المدفوع ونشرح لماذا هذا الشيء جيد. نفس المدخل نستخدمه لتشجيع العملاء الحاليين لترقية حساباتهم عندما يصلون إلى الحدود العليا من الخصائص المسموح باستخدامها في خطتهم الحالية.

العملاء الحاليون هم أفضل من تراهن عليهم في مبيعاتك. لا تخجل من تكرار عرض مبيعاتك على أشخاص يعرفون منتجك ويستخدمونه.

مصيدة الاسم

اجعل لتطبيقك اسم سهل التذكر

أكبر خطأ يرتكبه الكثير من الأشخاص اعتقادهم بأن اسم تطبيقهم يجب أن يصف التطبيق بشكل مفرط. لا تحرص على اختيار اسم يصف بوضوح الغرض من أداتك ،فهذا يؤدي في الغالب إلى اسم عام و معرض للنسيان. Basecamp اسم أفضل من Project Management Center أو ProjectExpress. و Writeboard أفضل من CollaborEdit .

مجموعات التركيز ولجان التسميات ستكلفك الكثير.لذا اختر اسم قصير و جذاب وقابل لتذكر ومن ثم أعمل به.

لا تنزعج إذ لم تحصل على اسم النطاق الذي تريده على الويب. تستطيع أن تكون خلاقاً وتحصل على ما تريد من خلال إضافة حرفين (مثال backpackit.com أو campfirenow.com) .

سهلة التطبيق

ألم تدرك الصناعة التقنية أن التفكير في أسماء جذابة و تصف المنتج بشكل واضح جدا سوف ينعكس على مصلحتها فقط؟ الشركات ستبيع المزيد من المنتجات مهما كانت هذه البرمجيات، لأنها لن تخشى من أن العملاء سيظلون معزولين عن عضوية نادي التقنية المتقدمة ، بسبب غطرسة المهندسين. على التقنيات أن تدرك ذلك أيضاً ، المنتج الجديد يجب أن يكون سهل الوصف وسهل الاستخدام و سهل في الشراء – وهذا بالنسبة للشركات يعني الأسهل في البيع.

—ديفيد باق، صحفي في، New York Times (from ما اسم المنتج؟)

الفصل الرابع عشر: الدعم

اشعر بالألم

أزل الحواجز بين الدعم الفني و فريق التطوير

في قطاع المطاعم، هناك اختلاف كبير بين من يعمل في المطبخ وبين من يتعامل مع العملاء. من المهم أن يفهم ويراعي كل طرف الآخر. لهذا نرى في غالبية مدارس الطبخ والمطاعم يكون طباخيها يعملون في الخارج كنادلين، لكي يتفاعل فريق الطبخ مع العملاء ويرى الأمور على حقيقتها.

الكثير من مطوري البرمجيات يحدث لهم نفس الشيء. فالمصممين و المطورين يعملون في "المطبخ" في حين يتولي الدعم الفني مهمة التعامل مع العملاء. لسوء الحظ هذا الشيء يعني أن المصممين والمطورين لن يستمعوا إلى الرغبات الحقيقية للعملاء. وهنا يحدث الإشكال لأن الاستماع إلى العملاء هو أفضل وسيلة لمعرفة مواطن القوة والضعف في المنتج.

ما الحل؟ تجنب عزل فريق التطوير/ التصميم عن عملائك. **لا تسند مهمة الدعم الفني إلى مركز اتصال أو إلى أطراف خارجية.** قم بها بنفسك. يجب أن تكون أنت وكامل فريق عملك على دراية تامة بما يقوله العملاء. عندما يتضايق عملاؤك يجب أن تعرف بذلك. أنت بحاجة لسماع شكواهم وأن تتضايق مثلهم.

في 37signals جميع رسائل الدعم الفني يُجاب عليها شخصياً من قبل العاملين الذين قاموا ببناء المنتج. لماذا؟ أولاً لأن هذا الشيء يقدم أفضل دعم للعملاء. ولكي يحصل العملاء على إجابة مباشرة من مطور التطبيق نفسه. أيضاً هذا الشيء يبقينا بقرب مستخدمي منتجاتنا والمشاكل التي يواجهونها. فعندما يصاب العملاء بالإحباط نحبط مثلهم. و نقول "أننا نشعر بك" بكل ما تحمله الكلمة من معنى.

من المغري الاعتماد على محلل إحصائي للتعرف على الأخطاء المتكررة. لكن الإحصائيات ليست كأصوات العملاء. لذا من الضروري التخلص قدر المستطاع من الحواجز التي بينك وبين أصوات العملاء الحقيقية .

الواجهة الأمامية لأي عمل هي ما يحدث فيها التفاعل النشط مع العملاء. قف هناك. أجعل المصممين والمطورين يعملون كنادلين. أقرأ رسائل العملاء، أستمع لاحباطاتهم واقتراحاتهم وتعلم منها.

تجنب استخدام الوسطاء

تقريباً كل أعمال التطوير و الدعم الفني و التسويق في Campaign Monitor تؤدى بواسطة شخصين. حتى لو اضطررنا إلى زيادة فريقنا، فلن نقوم بفصل الدعم عن التطوير. بالاستجابة الشخصية على كل طلب، نضع أنفسنا مكان العملاء و نجبر أنفسنا على أن ننظر بمنظورهم.

لا يكفي أن تعرف ماذا يريد العميل بل يجب أن تفهم لماذا يريده. هذا النوع من الفهم في الغالب يؤثر وبشكل مباشر على كيفية تصميم المنتج. تخلص من الوسطاء. من السهل جداً أن تعطي العملاء ما يحتاجونه عندما تكون قريباً منهم.

لقد ناقشت هذا الشيء مع أشخاص كثر وكانت إجابتي في الغالب على "أليس من الأفضل أن تعين موظفاً مبتدءاً ليتولى الدعم الفني؟" أن أقول : ضع نفسك مكان العملاء. إذا أردت شرائح اللحم المشوية بدرجة استواء معينة هل تفضل أن تتحدث إلى النادل أو إلى رئيس الطهاة الذي سيقوم بطهيها بنفسه؟

—ديفيد قراينر، مؤسس، Campaign Monitor

لا دورات تدريبية

استخدم المساعدة الداخلية و الأسئلة المتكررة كي لا يحتاج منتجك إلى دليل إرشادي أو تدريب

لا نحتاج إلى دليل إرشادي لاستخدام موقع ياهو أو جوجل أو أمازون. لذلك لماذا لا تقوم بعمل منتج لا يحتاج إلى دليل إرشادي؟ جاهد من أجل صنع أداة لا تحتاج أي تدريب للتعامل معها.

كيف تعمل ذلك؟ حسناً، كما أشرنا سابقاً، أجعل كل شيء سهلاً منذ البداية. كلما كان التطبيق سهلاً كلما تناقصت حاجة العملاء لمساعدتك. وبعد ذلك، أفضل طريقة لتقديم الدعم هي استخدام المساعدة الداخلية و الأسئلة المتكررة للإجابة على الأمور غير الواضحة.

على سبيل المثال، في تطبيقنا Basecamp نقدم الدعم للعملاء مباشرة من على الشاشة المخصصة لرفع صور شعاراتهم . بعض الأشخاص يواجهون مشكلة استمرار رؤيتهم للشعار القديم بسبب مسألة التخزين المؤقت الذي يقوم به المتصفح. لذا بجانب زر "أرفع شعارك" أضفنا رابط إلى الأسئلة المتكررة لنخبر العملاء بضرورة تحديث الصفحة ليروا الشعار الجديد. قبل وضعنا لهذا الرابط ، كنا نتلقى 5 رسائل يومياً بخصوص هذه المشكلة. أما الآن انتهينا منها.

إجابة سريعة

التعامل السريع مع الاستفسارات لابد أن يكون له الأولوية القصوى

يفرح العملاء عندما تجيب على استفساراتهم بسرعة. اعتاد العملاء على الإجابات المعلبة ، لذا من خلال الإجابات الفورية و المدروسة جيداً تستطيع أن تميز نفسك عن المنافسين. في 90 دقيقة وأحياناً في نصف ساعة من ساعات العمل نجيب على 90% من رسائل الاستفسارات التي تصلنا . والأشخاص فرحون جداً بهذا الشيء.

حتى وإن لم تكن لديك إجابة صحيحة 100% قل ما تستطيعه. تستطيع أن تملك ود الآخرين من خلال إجابة سريعة و واضحة وصريحة. إذا اشتكى شخص ما بشأن قضية لا تستطيع تصحيحها فوراً، أخبره "وصلنا ما قلته وسنعمل عليه في المستقبل" ،هذه طريقة ممتازة لامتصاص غضب العميل.

يقدر العملاء الصراحة وسيتغير موقفهم في الغالب ليكون أكثر لطفاً وأقل غضباً إذا أجبت عليهم بسرعة وبأسلوب واضح وصريح.

حشد كبير من العملاء

كيف يمكن لفريق مكون من ثلاث مطورين تقديم منتجات ناجحة ومبتكرة تنافس منتجات الشركات العملاقة؟ الإجابة تجنيد حشد كبير من الأشخاص.

تذكر من اليوم الأول أن العملاء أحد أهم الأصول لديك وأكثرها حيوية وتأثيراً على نجاحك في المدى الطويل، لذا عامل مجتمع المستخدمين كالملوك. طريق المنافسة مع الشركات الكبيرة يكون بالبدء بمشاريع صغيرة و الاهتمام بكل شخص من عملائك.

عملاؤك هم أول من سيلفت انتباهك إلى الأخطاء. وهم أول من سينبهك إلى الاحتياجات غير المشبعة. وهم أول من سيحمل العلم منطلقين لنشر رسالتك.

هذا لا يعني أن يكون منتجك مثالياً عند إطلاقه. بل على العكس تماماً، قم بالإطلاق المبكر، وعندما يواجه عملائك بعض المشاكل، تأكد من إرسال الرد إليهم وبسرعة وأشكرهم على إسهاماتهم.

لا يتوقع العملاء أن يكون منتجك مثالياً ولا يتوقعون كذلك أن كل ما يرغبون به من مميزات موجودة فيه. ولكن العملاء يتوقعون منك سماعهم وشكرهم، لذا أظهر لهم هذا الاهتمام. هذا أحد المجالات الذي تظهر فيه معظم الشركات الكبيرة عاجزة ، لذا عليك بالاهتمام به في مراحل مبكرة.

في موقع Blinklist كل رسالة عميل يُجاب عليها في غضون ساعة (ما لم نكن نائمين). كذلك لدينا منتدى على الإنترنت لنضمن أن كل موضوع و تعليق يأخذ نصيبه من الشكر.

وبنفس القدر من الأهمية، كل مطورينا يتلقون اقتراحات العملاء ويشاركون بنشاط في منتديات الحوار. بهذه الطريقة نحن نتحرك ببطء ولكن نبني مجتمعاً نشيطاً وموالياً لموقع BlinkList .

— مايكل رينق، أحد مؤسسي موقع، MindValley و Blinklist

المحبة الصارمة

كن مستعداً لقول لا لعملائك

عندما يتعلق الأمر بالمميزات المطلوبة، لا يكون العميل دائماً على حق. إذا أضفنا كل شيء طلبه عملائنا ، ففي الأخير لن يرغب أحد بمنتجاتنا.

لو أطعنا أهواء عملائنا، لكان تطبيق Basecamp عبارة عن : تتبع الوقت، نظام دفع فواتير شامل، وتنظيم الاجتماعات، إعداد جداول زمنية، نظام المهام التابعة لبعضها البعض، مراسلات فورية، وظائف الويكي وبشكل شامل، وكل ما تتخيله من أمور أخرى.

ومع ذلك فإن الأولية (1) التي حصلنا عليها من استبيانات العملاء تشير إلى أن يبقى تطبيق Basecamp بسيطاً .

إليك مثال أخر: بالرغم من بعض الشكاوي، قررنا أن لا ندعم متصفح IE5 في منتجاتنا. والذي كان يحتل نسبة 7% في السوق المستهدف. وقررنا أن النسبة الباقية 93% هي الأكثر أهمية والأجدر بالاهتمام. تصحيح الأخطاء وإجراء الاختبارات لدعم IE5 لا تستحق أن نصرف من أجلها الوقت. بدل من ذلك قررنا تقديم منتج أفضل للبقية الباقية.

كشركة تطوير برمجيات، يجب أن تكون كالفلتر. ليس كل ما يُقترح يطبق. علينا بالاهتمام بكل الطلبات لكن العميل ليس على حق دائماً. سيكون لديك وقت أكبر عندما تصرف بعض الأشخاص بعيداً عنك ، من الصعب القيام بهذا الشيء لكن هذه طبيعة العمل.

ومما له علاقة بهذا، من الأهمية بمكان لشركة تطوير البرمجيات أن تحب منتجاتها. و لن تحب منتجك إذا كان ممتلئاً بأمور لم تقتنع بها من الأساس. وهذا مبرر أخر للاعتراض على طلبات العملاء التي لا ترى أنها ضرورية.

في المنتدى الصحيح

استخدم المنتديات و المحادثات ليساعد العملاء بعضهم البعض

المنتديات و مجموعات المحادثات المتواجدة على الإنترنت هي وسيلة رائعة تسمح للعملاء بأن يطرحوا أسئلتهم ويساعدوا بعضهم البعض. عندما تتخلص من الوسيط ، ستوفر مسار من الاتصالات المفتوحة والمتدفقة وتوفر على نفسك الكثير من الوقت.

في منتديات منتجنا ، يضع العملاء المقالات والتلميحات و الحيل والمميزات المطلوبة والقصص وغيرها الكثير. نزور المنتدى من وقت لأخر لتقديم بعض المساعدة ، ونلاحظ أن المنتديات هي مكان رئيس للمجتمع لمساعدة بعضهم البعض وتبادل الخبرات حول المنتج.

سوف تتفاجأ من قدر رغبة الناس في مساعدة بعضهم البعض.

صرح بأخطائك

تخلص من الأخبار السيئة نهائياً

إذا حدث خطأ ما ، أخبر الناس عنه. حتى وإن لم يلاحظوه وقت حدوثه.

على سبيل المثال، تطبيقنا Basecamp سبق له أن توقف لعدة ساعات في منتصف الليل. رغم أن 99% من عملائنا لم يعرفوا بذلك، لكننا كتبنا في مدونتنا موضوع بعنوان "توقف غير متوقع" . نعم فعملائنا يستحقون معرفة ذلك.

وهذه عينة مما كتبناه عندما حدثت المشكلة: "نعتذر عن التوقف الذي حصل صباح هذا اليوم- كانت لدينا بعض المشاكل في قاعدة البيانات والتي تسببت في البطء والتوقف لبعض العملاء. قمنا بإصلاح المشكلة واتخذنا احتياطاتنا اللازمة لضمان عدم حدوث هذا الشيء مرة أخرى.. شكراً جزيلاً على انتظارك، وشكراً مرة أخرى، ونأسف لهذا التوقف."

كن منفتحاً و نزيهاً وشفافاً قدر استطاعتك. لا تحتفظ بالأسرار أو تخفيها. العميل العارف بالتفاصيل هو أفضل عميل. بالإضافة إلى ذلك، ستدرك أن معظم أخطائك لم تكن سيئة في أذهان عملائك كما كنت تتصور. العملاء في الغالب سيكونون سعداء وسيمنحونك فرصة لاستعادة وضعك ، طالما أنك كنت صادقاً معهم.

ملاحظة جانبية بخصوص كيفية إعلام العملاء بالأخبار: عندما يكون هناك أخبار سيئة اعرضها دفعة واحدة وبكل صراحة. أما الأخبار السارة

فأرسلها لهم ببطء. فإن كنت تستطيع إطالة فترة سعادتهم ففعل ذلك.

كن سريعا ً و مباشرا ً و صادقا ً

قد يبدو غريباً عندما نقول أن أفضل سيناريو عندما تكتب الشركة عن نفسها تقارير سيئة. هذا الشيء يجعل الشركة مترقبة ويمنعها من أن تكون في وضع ضعيف أو في موقف دفاعي.

—قرج شاروين، نائب رئيس التطبيقات التكنولوجية، CNET، و ايميلي أفلا، رئيسة، Calypso Communications (المصدر تمهيد لأزمة العلاقات العامة)

الفصل الخامس عشر: قبل الإنطلاق

تجربة تمهيدية لمدة شهر

تحديث الأوضاع الرئيسية في المنتج بعد 30 يوماً من الإطلاق

تحديث سريع يُظهر قوة المنتج الدافعة، يُظهر انك تستمع، يُظهر انه مازال لديك مزيد من الحيل والأوراق لتكشفها، ويوفر لك موجة ثانية من الإبداع حيث يؤكد من جديد مشاعر أولية جيدة نحوك ويعطيك شيئاً لتتحدث عنه وللآخرين للتدوين عنه.

أيضاً الترقية السريعة تجعلك تُركز على المكونات الأكثر حسما قبل إطلاق البرنامج. و بدلاً من محاولة العصر في بعض الأشياء، من الأفضل أن تبدأ في إتقان خاصية المنتج الأساسية ثم التحليق بالمنتج في العالم الحقيقي، عندما يخرج هناك تستطيع الحصول على تعليقات العميل و ستعرف أي جزء يتطلب انتباه مستقبلي .

هذه نظرة قريبه من خطوة الطفل التي عملت جيداً مع منتج Backpack أطلقنا المنتج الأساسي أولاً ثم، بعد بضعة أسابيع، أضفنا خصائص مثل جوال Backpack للحاسبات المحمولة باليد ومنذ ذلك الحين اخبرنا عملائنا أنها أكثر الخصائص رغبة لديهم!

دعْ التدوينات تأتي

اظهر أن منتجك على قيد الحياة بإبقائه في تطوير مستمر وتحديث مستمر للمدونة بعد الإطلاق.

لا تتوقف عن التدوين بمجرد الإطلاق، ابرز أن منتجك كائن حي بإبقاء المدونة في تحديثات مستمرة (على الأقل مرة واحدة في الأسبوع، أو أكثر كلما استطعت) .

بأمور تشمل:

- أسئلة شائعة وأجوبتها.
- كيف تعمل.
- نصائح وحيل.
- خصائص جديدة، تحديثات، وإصلاحات.
- انتشار المنتج/ تعليق الإعلام على المنتج.

المدونة لا تُظهر فقط أن تطبيقك على قيد الحياة فقط؛ بل تجعل شركتك تبدو أكثر إنسانية. مرة أخرى، لا تقلق من إبقاء اللهجة صديقة وإنسانية (أي ودوده). بعض المجموعات الصغيرة تشعر أحياناً أنهم بحاجة إلى لهجات ضخمة أو أكثر احترافية ومهنية طوال الوقت، هذا يشبه تقريباً النسخة التجارية من تعقيد نابليون! أيضاً لا تتحدث بضعف. امرح في الحقيقة فأنت تستطيع التحدث مع العميل كما لو كان صديقاً.

انه على قيد الحياة

تحديث مدونة المنتج باستمرار هو أفضل مؤشر أن تطبيق الويب في حالة تطوير نشطة، هذا يكون محبباً فيه و يدل على أن هناك ضوءاً في المنزل. مدونة المنتج المهجورة هي دليل على منتج مهجور وتتجه أحاديث المهتمين بتهمة النوم العميق للقائمين على المنتج.

احتفظ بالمحادثات مع عملائك على مدونة منتجك، وكنْ شفافاً وسخياً بالمعلومات التي تشاركها. دعْ شركتك تتألق من خلال فلسفتها، منفتحة وتناقش المنافسين. اشرْ إلى خصائص قادمة وعلّق باستمرار على الآراء الراجعة.

المنتج الحي هو الذي يتحدث ويستمع إلى مستخدميه، تحديث مدونة المنتج المتعاقبة يعزز الشفافية وإحساس الانتماء للمجتمع والولاء لعلامتك التجارية ثم إن الدعاية الإعلانية هي المكافأة.

كما قال محرر في Lifehacker، افحص مدونات المنتج لتطبيقات الويب التي أحبها – مثل قووقل، فليكر، ياهوو، del.icio.us، و 37signals . أنا أكثر ميلاً لذكرها والكتابة عنها من أن ابعث بجانب واحد لإصدارات تطبيقات الويب إلى الصحافة وهي لا تحافظ على محادثة مفتوحة مع مستخدميها وجماهيرها.

Gina Trapani, web developer and editor ofLifehacker, the productivity and software guide—

الأفضل، ليس بيتا

لا تستخدم النسخة التجريبية كـ كبش فداء

هذه الأيام تُشعرك بان كل شيء في مرحلة تجريبية للأبد، هذه قمة التوقف. المرحلة التجريبية اللا منتهية تخبر العملاء: انتم غير ملزمون حقاً باستخدام النسخة النهائية من المنتج، هي تقول: "استخدم هذه النسخة، ولكنها ليست ممتازة، انه ليس ذنبنا. "

النسخة التجريبية تمرر الإحباط لعملائك ومستخدمي المنتج، إن لم تكن واثقاً كفاية حول إصدارك فماذا تتوقع من الجمهور أن يكون؟ نسخة تجريبه سرية أمر جيد، نسخة تجريبية للجمهور هو هراء! إذا لم تكن جيدةًً كفاية لاستخدام الجمهور فلا تطلقها لاستنفاذ صبر الجمهور.

لا تنتظر منتجك أن يصل إلى الكمال، فلن يحصل هذا. تحمل مسئولية ما ستقوم بإصداره. ضعه في العالم الحقيقي وسمي ذلك إصداراً. بخلاف ذلك، أنت تصطنع الأعذار فقط.

النسخة التجريبية لا معنى لها

يُلام قووقل و آخرون، لتسببهم بمشاكل مثل هذه. الآن، تم تدريب المستخدمين بواسطة عدد من الخبراء على أن "النسخة التجريبية" لا تعني في الحقيقة أي شيء.

Mary Hodder, information architect and interaction designer (from The Definition of Beta)—

في كل الوقت

هل أنا فقط، أم أننا جميعاً في نسخة تجريبية، كل الوقت ؟

—Jim Coudal, founder, Coudal Partners

لا تحدّث جميع الثغرات بشكل متساو

حدد أولويات ثغراتك (حتى لو تجاهلت بعضها)

فقط لأنك اكتشفت ثغرة في منتجك، لا يعني أن الوقت حان للذعر. جميع البرمجيات تمتلك ثغرات – إنها مجرد حقيقة من حقائق الحياة.

ليس عليك أن تصلح كل ثغرة فورياً، معظم الثغرات المزعجة ليست هدّامة. المزعجة منها يمكن جدولتها قليلاً والثغرات التي تؤدي إلى أخطاء مثل "انه لا يظهر بشكل صحيح" أو أخطاء أخرى يمكن أن تظل بسلام لبعض الوقت لكن من الواضح ان الثغرة إذا دمرت قاعدة بياناتك فالواجب عليك إصلاحها حالاً.

حدد أولويات ثغراتك، كم شخصاً قد تضرر؟ ما مدى سوء المشكلة؟ هل تستدعي هذه المشكلة انتباه فوري أو يمكنها الانتظار؟ ما يمكنك عمله الآن سيكون له اكبر الأثر في أكبر عدد من الأشخاص؟ كثيراً من المرات إضافة خاصية جديدة تكون أكثر أهمية لتطبيقك من إصلاح ثغرة موجودة.

أيضاً، لا تنشئ لديك هالة الخوف من الثغرات المحيطة، الثغرات تحصل. لا تحاول باستمرار أن توجه اللوم على احدهم. آخر شيء تريده هو بيئة تُخفي الثغرات ببطانة سميكة عوضاً عن النقاش المفتوح.

وتذكر ما قلناه مبكراً حول أهمية الصدق، إذا كان العملاء يشتكون من ثغرة فكن واضحاً معهم. اخبرهم انك لاحظت هذا الخلل وتتعامل معه. إذا لم تكن مواجهاً على الفور، اشرح لماذا؟ وقل انك تركز على أجزاء في المنتج تؤثر على عدد كبير من الأشخاص. الصدق هو أفضل سياسة.

امتطاء العاصفة

انتظار ردود الأفعال غير المحسوبة للتغييرات تخمد قبل اتخاذ أي إجراء

عندما تُأرجح القارب، سيكون هناك أمواج. عندما تقوم بتعريف خاصية جديدة، أو تغيير الشروط، أو إزالة شيئاً ما، ردود الفعل غير المحسوبة تكون غالباً سلبية؛ وسوف تتدفق في قاربك.

مقاومة الرغبة في الذعر، أو تغيير الأمور بسرعة في استجابةٍ للأوضاع. تُكون رغبة في الانفجار عند البداية ولكن إذا تخلصت من فترة الـ 24-48 الساعة الأولية، فان الأمور عادةً تهدأ. معظم الأشخاص يستجيبون قبل أن يفكرون ويستخدمون أي أمر أضفته (أو أي أمر أزلته) وينتقدونه. لذا استرح، واتركها جميعاً، لا تقم بالإزالة إلا بعد مرور بعض الوقت. بعد ذلك سيمكنك تقديم أكثر من تفسير للاستجابة.

أيضاً، تذكر أن ردود الأفعال السلبية تكاد تكون دائما أعلى الأصوات وأكثر عاطفية من الايجابية منها. الحقيقة، انك قد تكون سمعت الأصوات السلبية فقط حتى لو كانت غالبية قاعدتك سعيدة بهذا التغيير. تأكد من انك لا تتراجع بحماقة في الأزمة ولكن جادل ثم قرر.

مواكبة الجيران

اشترك في خلاصة إخبار منافسوك

اشترك في خلاصات إخبارية حول كلاً من منتجك ومنتجات منافسيك (من الحكمة دائماً معرفة الطريق لأحد أعداءك). استخدم خدمات مثل Feedster ،Technorati ،PubSub، وآخرون للبقاء على تحديث يومي (ابحث عن الكلمات الرئيسية، استخدم أسماء الشركة وأسماء المنتج). مع RSS، معلومات التغيير المستمرة ستصلك فوراً لذا أنت تعلم دائماً بالجديد وبأسرع وقت.

احذر من انتفاخ المسخ!

أكثر نضجاً لا يعني بالضرورة أكثر تعقيداً

لا تتهاون في مقاومة انتفاخ المنتج كلما تقدمت الأمور. الإغراء في التطوير سيكون لرفع المستوى، ولكن ليس بالضرورة أن يكون بهذه الطريقة. كون الشيء أصبح أكبر وأكثر نضجا لا يعني بالضرورة أن يصبح أكثر تعقيداً.

لا يجب عليك أن تكون قلما في الفضاء الخارجي تكتب رأساً على عقب. لا بأس أن تكون في بعض الأحيان قلم رصاص فقط (كناية عن الفرق بين التركيز في المشكلة والتركيز على حل المشكلة). كما انك لست في حاجة لان يكون لديك سكين الجيش السويسري. يمكنك أن تكون مفك براغي، ولا تحتاج لبناء مراقبة مغطس الماء الذي يكون آمن عند 5000 متر إذا عملائك من عشاق الأرض ويريدون معرفة ما هو الوقت.

لا تتضخم لأجل الحصول التضخيم فقط. هذه هي كيفية انتفاخ التطبيقات.

الجديد لا يعنى دائماً تحسين. يوجد أحيانا نقطة يجب عليك أن تترك المنتج ليفعلها.

هذه واحدة من الفوائد الرئيسية لبناء برمجيات تعتمد على شبكة الانترنت بدلاً من البرمجيات التقليدية المعتمدة على سطح المكتب. صانعي برمجيات سطح المكتب مثل Microsoft ،Intuit ،Adobe يحتاجون لبيعك نسخاً جديدة كل عام، ونظراً لعدم استطاعتهم بيعك نفس النسخة؛ عليهم تبرير تكاليف النسخ الجديدة بإضافة خصائص جديدة وهنا يبدأ انتفاخ المنتج.

مع برمجيات تعتمد على الويب فهي تركز في الأساس على نموذج الاشتراكات، يدفع الأشخاص شهرياً مقابل استخدام الخدمة.. أنت لست بحاجة إلى للإبقاء على بيعهم بإضافة المزيد والمزيد والمزيد، تحتاج فقط إلى تقديم خدمة ثمينة متطورة.

اتبع التدفق

كنْ منفتحاً لمسارات جديدة و تغييرات جديدة في الاتجاه.

الجزء الجميل في تطبيقات الويب هي السلاسة والمرونة. لا تقيده في صندوق، اشحنه، وانتظر سنوات حتى الإصدار القادم. يمكنك التغيير كلما استمر البرنامج. كن منفتحاً للحقيقة أن فكرتك الاعتيادية قد لا تكون أفضل فكرة.

انظر لفليكر، تبدو كلعبة مباشرة بعدة لاعبين تُسمى لعبة اللانهاية. سرعان ما أدرك صانعوها أن حقيقة تشارك الصور في اللعبة هي أكثر المنتجات قبولاً من اللعبة نفسها (وكانت في آخر الأمر مصروفة من الخدمة) . كُنْ مستعداً لقبول الأخطاء وتغيير المسار.

كن راكب أمواج، شاهد المحيط. حدد أين تنكسر الأمواج الكبيرة وعدّل مسارك وفقاً لذلك.

الفصل السادس عشر: الختام

الآن أنطلق

انتهينا!

حسناً ، قرأت الكتاب! نأمل أن تكون تأثرت به لتبدأ بإيصال تطبيقك إلى الواقعية. في السابق لم يكن من الممكن تقديم برامج رائعة في ظل نقص الموارد. مع توفر الأفكار والحماس و الوقت والمهارات اللازمة لا حدود لما يمكن إنجازه.

بعض الأفكار الختامية:

التنفيذ

كل شخص يستطيع قراءة كتاب. و يستطيع أن يأتي بفكرة. كل شخص لديه قريب بوظيفة مصمم ويب. كل شخص يستطيع أن يكتب في مدونة. كل شخص يستطيع استئجار شخص ما ليساعده في كتابة بعض الأكواد.

سيكون الاختلاف بينك وبين الآخرين هو مدى جودة التنفيذ. فالنجاح قائم على التنفيذ الجيد.

في البرمجيات، هذا يعني القيام بكثير من الأشياء بطريقة صحيحة. الكتابة الرائعة لا تكفي إذا تبعها إخفاق في تنفيذ وعود تلك المقطوعة النثرية. تصميم واجهة واضحة لا يمكن إنجازه بشكل جيد إذا كان كودك ممتلئ بالأخطاء. التطبيق الرائع سيكون عديم الجدوى إذا كانت طرق الترويج لا يعرف عنها أحد. لتحقيق نتائج أفضل ، عليك أن تجمع كل هذه العناصر.

السبيل إلى ذلك هو التوازن. إذا انحرفت بعيداً عن أحد المسارات، أعلم إنك اتجهت إلى الفشل. أبحث باستمرار عن نقاط الضعف ثم قم بتصحيحها.

الأشخاص

أمر يستحق إعادة التأكيد ، الشيء الوحيد الذي نعتقد أنه أهم عنصر في بناء تطبيق ناجح : الأشخاص العاملين. التعويذات Mantras، التصميم البؤري، البرمجيات الأقل وغيرها من أفكار رائعة ،غير مهمة إذا لم يكن في فريقك الأشخاص المناسبين لتنفيذها.

أنت بحاجة إلى أشخاص شغوفين بما يقومون به. أشخاص يهتمون بأعمالهم – ويفكرون بها فعلا على أنها مهنة. الأشخاص الفخورين بعملهم، بغض النظر عن العوائد المادية التي يحصلون عليها. الأشخاص الذين يعملون بجد على أدق التفاصيل حتى ولو كان 95% من المتعاملين مع التطبيق لن يلاحظوا الفرق. الأشخاص الذين يرغبون ببناء شيء عظيم دون الاستسلام لنقص الموارد. الأشخاص الذين يرغبون بالعمل الجماعي.على أي حال عندما تجد هؤلاء الأشخاص تمسك بهم. في الختام، الأشخاص في فريقك قد يبنون أو يهدمون مشروعك وبالتالي شركتك.

أكثر من مجرد برمجيات

من الخطأ الاعتقاد بأن مفهوم الوصول للواقعية ينطبق فقط على بناء تطبيق ويب. فور ما تبدأ بتأمل الأفكار المحيطة، سترى هذا المفهوم في كل مكان. بعض الأمثلة على ذلك:

- قوات العمليات الخاصة، مثل قرين باريتس أو نيفي سييل ، تستخدم فرق صغيرة وتعتمد على الانتشار السريع لإنجاز المهام التي تعجز عنها الوحدات الأخرى بسبب كبر حجم أو بطء تلك الوحدات.
- ذا وايت سترايبس –فرقة موسيقية- قضت على المعوقات من خلال التزامها بالأمور البسيطة : شخصين ،أغاني متناغمة، عزف هادئ، تقليص وقت التسجل إلى الحد الأدنى إلى غيره.

- جهاز آي بود من أبل ميز نفسه عن المنافسين بخلوه من بعض المميزات مثل الراديو أو مسجل الصوت.
- بإتباع إستراتيجية تقليل الوقت يتمكن اللاعب في كرة القدم الأمريكية من قطع مسافة كبيرة.
- راتشيل ري أساس كتبها الناجحة المتخصصة في الطبخ وعروضها التلفزيونية هو اعتمادها على فكرة 30 دقيقة "أحصل على وجبة حقيقية".
- ارنيست همنغوي و ريموند كارفر استخدما لغة بسيطة و واضحة مازالت حتى الآن تؤثر في المتلقي.
- شكسبير لعب في القيود المفروضة على السونيت - القصائد الغنائية ذات الأربعة عشر بيتاً في قصائد الأيامبك خماسية التفاعيل.
- وغير ذلك الكثير...

بالتأكيد الوصول للواقعية يدور حول بناء برمجيات رائعة. لا يوجد سبب لاكتفائك بهذا القدر . خذ هذه الأفكار وحاول تطبيقها في شتى المواقف في حياتك. من المحتمل إنك ستحقق نتائج رائعة.

أبق على اتصال

دعنا نعرف كيف أثر بك كتاب "الوصول للواقعية". أرسل إيميل إلى gettingreal [at] 37signals [dot] com.

بالإضافة كن مطلعاً على أخر التحديثات في 37signals من خلال زيارة Signal vs. Noise، وهي مدونتنا تتحدث عن الوصول للواقعية، و قابلية الاستخدام، والتصميم، وغيرها من المواضيع.

شكرا على القراءة، وحظاً سعيداً!

مصادر 37signals

- موقع 37signals
- مدونة Signal vs. Noise
- Basecamp— تطبيق ويب للمشاريع التعاونية .
- campfire — تطبيق ويب لدردشة فريق العمل.
- Backpack— تطبيق ويب منظم للمعلومات .
- Writeboard— تطبيق ويب للكتابة التعاونية .
- Ta-dalist— تطبيق ويب لعمل قوائم مهام الأعمال اليومية بسيط جداً .
- Ruby on Rails— إطار عمل مفتوح المصدر لبرمجة تطبيقات الويب

الترجمة

كل الشكر لـ محمد الرحيلي، سلوى القاسمي، رشا الأحمد

The rest of the book is not yet translated into this language. Contact us via email if you'd like to help translate. The entire book is available in English.

Getting Real Overview